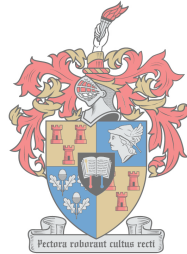


Infrasoniese, GPS-gesinkroniseerde klankopnemers vir die opsporing van olifantdreungeluide

deur

Rudolf Willem Byker



UNIVERSITEIT
iYUNIVESITHI
STELLENBOSCH
UNIVERSITY

*Tesis ingelewer ter gedeeltelike voldoening aan die vereistes vir die graad
Magister in Elektroniese Ingenieurswese in die Fakulteit Ingenieurswese
aan die Universiteit van Stellenbosch*

1918 - 2018

Studieleier: Prof. T.R. Niesler

Desember 2018



Verklaring

Deur hierdie tesis elektronies in te lewer, verklaar ek dat die geheel van die werk hierin vervat, my eie, oorspronklike werk is, dat ek die alleenouteur daarvan is (behalwe in die mate uitdruklik anders aangedui), dat reproduksie en publikasie daarvan deur die Universiteit van Stellenbosch nie derdepartyregte sal skend nie en dat ek dit nie vantevore, in die geheel of gedeeltelik, ter verkryging van enige kwalifikasie aangebied het nie.

Datum: Desember 2018

Kopiereg © 2018 Universiteit van Stellenbosch
Alle regte voorbehou.

Uittreksel

Die outomatiese opsporing van wilde olifante in hul natuurlike habitat, sonder enige interaksie met die dier self, het verskeie toepassings in wildbewing en bewaringsekologie. Huidige oplossings vereis dat apparate aan individuele olifante vasgemaak word. Olifantdreungeluide het sterk infrasoniese komponente en olifanthabitat bied ideale toestande vir die propagering van hierdie geluide oor lang afstande. 'n Geïntegreerde hardwarestelsel vir die opsporing van hierdie geluide word ontwerp en getoets. Dit bestaan uit klankopnemers wat betreklik goedkoop is, sensitief is vir lae frekwensies, battery-aangedrewe is, lae intrinsieke ruis het, draadloos kan kommunikeer, en met behulp van **globale posisioneringstelsel (GPS)** met mekaar gesinkroniseer is. 'n Eie klankopnemer is ontwerp omdat geen kommersieel beskikbare toerusting aan die vereistes van die projek voldoen nie.

Die klankopnemer gebruik 'n 12 V verseelde loodsuurbattery. Pasgemaakte kragbronne is ontwerp om die onderskeie dele van die opnemer aan te dryf. 'n Sirkelvormige skikking van 6 **Invensense ICS-40300 mikro-elektromeganiese sisteem (MEMS)**-mikrofone word op elke klankopnemer gebruik. 'n Pasgemaakte oudiokoppelvlak, wat uit versyferaars, voorversterkers en **teen-oorvleuelingsfilters** bestaan, is ontwerp. Elke oudiokoppelvlak het 'n hoë **intree-impedansie**, 'n hoogafsnijfrekwensie van ongeveer 1 kHz, 'n monsterbreedte van 24 bisse per kanaal, 'n monstrefrekwensie van 4800 Hz, en 6 onafhanklike kanale wat sinkroon gemonster word. 'n Digitale oudiobuffer versamel monsters vanaf die versyferaar en stuur dit via **serie-koppelvlak vir randapparatuur (SPI, Serial Peripheral Interface)** na 'n **Raspberry PI (RPI)** waar dit op 'n **Secure Digital (SD)**-kaart gestoor word vir latere verwerking. Ruimte word gelaat vir **intydse** verwerking op die **RPI** in die toekoms.

GPS-modules word gebruik om 'n akkurate tydstempel aan elke monster toe te ken sodat die oudiokanale van verskeie onafhanklike klankopnemers as 'n wyd-gespasieerde mikrofoonskikking gebruik kan word, sonder direkte kommunikasie tussen die opnemers.

Verskillende algoritmes vir die opsporing van klankbronne is gesimuleer en getoets. Veelsymetring met wyd-gespasieerde mikrofone is ondersoek, en daar is bevind dat ten minste vier wyd-gespasieerde klankopnemers met 'n enkele mikrofoon op elkeen nodig is om akkurate klankbronposisies te bepaal. 'n Nuwe algoritme vir klankbronrigtingberaming, wat op die tydverskilberamings van elkeen van die 15 mikrofoonpare in die 6-mikrofoon skikking staatmaak, word ontwikkel, gesimuleer en getoets. Hierdie algoritme het beter resultate gebied as veelsymetring wanneer slegs twee wyd-gespasieerde opnemers gebruik word. In beide gevalle is toetse uitgevoer met 'n luidspreker wat olifantdreungeluide of witruis speel. Figure 10.22–10.27 toon die resultate.

Die opsporing van olifantdreungeluide met infrasoniese klankopnemers blyk 'n haalbare oplossing vir verskeie wildbewaringsprobleme te wees.

Abstract

The automatic localisation of wild elephants in their natural habitat, without interaction with the animals themselves, has many applications in wildlife conservation and conservation ecology. Current solutions require the attachment of devices to individual animals. Elephant rumbles have strong infrasonic components and elephant habitats offer ideal conditions for the propagation of these rumbles over long distances. An integrated hardware solution for the localisation of these sounds is designed and tested. It consists of sound recorders which are relatively inexpensive, are sensitive to low frequencies, are battery-powered, have low intrinsic noise, can communicate wirelessly, and are synchronised with one another via **GPS**. A custom recorder is designed since no commercially available audio equipment meets these requirements.

The recorder uses a 12 V sealed lead acid battery. Power supplies are designed to power the respective parts of the recorder. A circular array of 6 **Invensense ICS-40300 MEMS** microphones are used with each recorder. An audio interface is designed, consisting of ADCs, preamps and anti-alias filters. Each audio interface has a high input impedance, a high-cutoff frequency of about 1 kHz, a bit depth of 24 bits per channel, a sample rate of 4800 Hz and 6 independent, synchronously sampled audio channels. A digital audio buffer collects samples from the **ADC** and sends it to a **RPI** over **SPI**, where it is stored on an **SD** card for later processing. In the future, processing may be done online on the **RPI**.

GPS modules are used to accurately timestamp each sample so that the audio channels from independent recorders may be used as a wide-spaced array of microphones without explicit inter-device communication.

Various sound source localisation algorithms are simulated and tested. Multilateration with wide-spaced microphones is investigated, and the conclusion is that at least four wide-spaced sound recorders with a single microphone on each is needed to accurately locate sound sources. A novel direction of arrival estimation algorithm based on the pairwise time delay estimations of all 15 microphone pairs in the 6-microphone array is developed, simulated and tested. This algorithm performed better than multilateration when only two wide-spaced sound recorders were used. In both cases, tests were performed with a loudspeaker playing elephant rumbles or white noise. The results are shown in Figures 10.22–10.27.

Suggestions are made for future work towards an end product that can be operated by park rangers and ecologists. The localisation of elephant rumbles with infrasonic sound recorders appears to be a viable solution to various wildlife conservation problems.

Dankbetuiging

- Almal op die RhinoNet span.
- Simon Smith en sy gesin by Louisvale wines, vir hul gasvryheid en die gebruik van hul wingerde vir oudiotoetse.
- Dirk Streicher en sy gesin, in wie se vakansiehuis dele van hierdie tesis geskryf is.
- Hankyu Kim, vir sy raad met stroombaanuitleg vir **MEMS**-mikrofone.
- Corrie Luijk, vir die gratis mikrofoon-windbeskermers (Figuur 4.9(b)).
- Phillip Strydom, vir die gebruik van sy ekstra rekenaar vir simulaties.
- Andries Stephanus Lourens Heystek, Jürgen Lüdemann, Etienne Neethling en Jonathan Wotherspoon, vir hul hulp by praktiese toetse.
- Johan Arendse, Wessel Croukamp en Wynand van Eeden, vir hul tegniese hulp met masjinerie, soldeerwerk, die 3D-drukker en die vervaardiging van stroombaanbordjies.
- Jenny Martin, vir haar geduldige en knap hantering van bestellings.
- Helmy Byker en Ingrid Hagg, vir hul ongeëwenaarde taalvernuf.
- Bertus Byker, vir huisvesting, 'n kar, opvoeding, 'n rotsvaste geloof, en nog baie meer as wat hier genoem kan word.
- Professor T.R. Niesler, vir die opregtheid waarmee hy sy werk aanpak.
- Belangriker nog: *God sy dank wat ons altyd laat triomfeer in Christus en wat die reuk van sy kennis deur ons oral versprei. Want ons is 'n aangename geur van Christus tot eer van God onder die wat gered word en onder die wat verlore gaan; vir die laaste 'n reuk van die dood tot die dood, maar vir die eerste 'n reuk van die lewe tot die lewe. En wie is tot hierdie dinge bekwaam?*
— 2 Korinthiërs 2:14–16 (Afr1953)

Woordelys

Engelse terme verskyn in hakies.

aaneengeskakelde lys (linked list) 'n Skikking waar die geheue vir elke element dinamies toegeken word en elke element 'n wyser stoor na die posisie van die volgende element in die geheue. 64

afbeelding (image) 'n Presiese duplikaat van 'n stoormedium (soos 'n SD-kaart of 'n hardeskyf) wat in 'n lêer op 'n ander medium gestoor word. 93, 169, 170, 173, 177, 181

afskop (boot) Laai 'n bedryfstelsel en begin om dit uit te voer. 74, 178

afstandsverskil (range difference) Die verskil tussen die afstand van 'n klankbron na een mikrofoon en die afstand van dieselfde klankbron na 'n ander mikrofoon. 130, 131

afvlakkapasitor (bypass capacitor) 'n Kapasitor met goeie werkverrigting by hoë frekwensies, wat gebruik word om toevoerspanningsruis te onderdruk. 16, 21, 28, 47, 151

apodisasie (apodisation) Die verswakking van 'n gevensterde sein naby die rande van die venster. Dit word ook soms 'venstering' genoem, maar ter wille van duidelikheid word daar in hierdie dokument onderskeid getref tussen 'venstering' en 'apodisasie'. Die woord beteken letterlik: 'verwydering van die voet'. Dit word gebruik om die amplitude van sylobbe in die frekwensiegebied te beperk wanneer die DFT van 'n sein met 'n eindige lengte geneem word. 140

bedryfstyd (uptime) Die tydsverloop sedert 'n stelsel aangeskakel is. 97

definisieversameling (domain) Die versameling van argumente waarvoor 'n funksie of sein gedefinieer is. 114

deurlaatband (pass band) Die frekwensie-omvang waarin 'n filter 'n sein deurlaat, oftewel nie verswak nie, relatief tot ander frekwensies. 29, 41–44

dienssiklus (duty cycle) Die gemiddelde persentasie tyd waarvoor 'n pulstrein 'n hoë waarde aanneem. 12, 96, 159, 160

draad (thread) Instruksies, ondergeskik aan 'n proses, wat onafhanklik van ander instruksies op 'n verwerker uitgevoer word, maar wat geheue deel met ander drade wat aan dieselfde proses behoort. 64–66, 70, 89, 95, 150

eentoppig (unimodal) Beskrywend van 'n waarskynlikheidsverdeling wat slegs een modus of top het. 125–127

ewekansig (uniformly distributed) Beskrywend van 'n toevalsveranderlike met 'n waarskynlikheidsverdeling wat konstant is oor die definisieversameling van die veranderlike. 126–128

- ferrietkern** (ferrite bead) 'n Komponent wat hoë-frekwensie ruis in 'n kabel of stroombaan-spoor onderdruk. 19, 21, 151
- gebibber** (jitter) Die mate van afwyking van 'n vermoedelik periodiese sein van sy nominale periode. 100
- grootkantig** (big-endian) Beskrywend van 'n getal waar die mees beduidende greep by die laagste geheue-adres gestoor word. 77
- haal** (pop) Haal die oudste element uit 'n waglys. 63–66, 70, 82
- ineenstort** (crash) Ophou werk (m.b.t. sagteware) 88, 90
- inset** (input) Die data, parameters, seine of strome wat 'n stelsel betree. 12–14, 38, 42, 44, 86, 117, 126, 158–160
- intree** (input) 'n Poort, posisie of kontakpunt waar 'n stelsel betree word. iii, 6–13, 15, 18, 20, 36–39, 41, 44, 45, 47, 50, 56, 57, 66, 69, 73, 88, 89, 93, 97, 149, 156, 161
- intyds** (realtime) Beskrywend van 'n proses of stelsel wat werk binne 'n sekere spertyd moet voltooi. iii, 66, 72, 83, 86, 93, 94, 150, 152, 153
- isotropies** (isotropic) Dieselfde in alle rigtings. 31
- kantigheid** (endianness) 'n Eienskap van 'n getal in rekenaargeheue wat bepaal watter kant van die getal (groot of klein) eerste gestoor word. 77, 82
- keerband** (stop band) Die frekwensie-omvang waarin 'n filter 'n sein stop, oftewel noemenswaardig verswak relatief tot ander frekwensies. 41, 43
- kerndigtheidsberaming** (kernel density estimation) Die vouprodukt van 'n versameling diskrete observasies met 'n waarskynlikheidsverdeling (die kern), om die verdeling waaruit die observasies geneem is te probeer beraam. 116, 118, 120, 124, 125, 129, 141, 143, 145–147
- kleinkantig** (little-endian) Beskrywend van 'n getal waar die mins beduidende greep by die laagste geheue-adres gestoor word. 82
- kleinstekwadratemetode** (method of least squares) 'n Metode waarvolgens 'n beste passing vir 'n oorgedetermineerde stel vergelykings bepaal kan word. 122, 124, 131–135
- konteksoorskakeling** (context switching) Die berging van die toestand (oftewel konteks) van 'n proses of draad, sodat uitvoering daarvan later hervat kan word. 65, 68
- lêergids** (filesystem) 'n Datastruktuur soos NTFS of ext4 wat die lêers in 'n partisie organiseer. 75, 170, 173, 177, 178
- lêeroorsprong** (root) Die pad op 'n Linux-stelsel wat '/' genoem word, wat dien as verwysingspunt vir die ligging van alle lêers op die stelsel. 75
- matriksomkering** (matrix inversion) Die berekening van die inverse van 'n matriks. 116, 133, 135
- meertoppig** (multi-modal) Beskrywend van 'n waarskynlikheidsverdeling wat meer as een modus of top het. 126, 127

- mikrofoonpaar** (microphone pair) Twee mikrofone wat saam as een sensor gebruik word. Dit is die eenvoudigste moontlike mikrofoonskikking. 112, 113, 119, 120, 122, 124, 125
- mikrofoonskikking** (microphone array) Twee of meer mikrofone wat saam as een sensor gebruik word. 7, 121, 122, 124
- monstertempo** (sampling frequency) Die aantal monsters wat per tydeenheid van 'n sein geneem word. 19, 22, 27, 36, 38, 41, 51, 65, 96
- monsteruitdunning** (downsampling, decimation) Die vermindering van monsters om die monstertempo te verlaag. 119, 120
- multipadversteuring** (multipath interference) Versteuring wat veroorsaak word deurdat 'n sein langs meer as een pad 'n sensor bereik, terwyl die vertraging langs die roetes verskil. 100
- nabye veld** (near field) Die gebied naby 'n sensorskikking waar die kromming van die golf front 'n noemenswaardige invloed op die gemete seine het. 112, 121, 122, 130
- nulstopping** (zero padding) Die byvoeging van nulle aan die begin of einde van 'n sein, of tussen monsters. 115
- onderbrekingsprioriteitsvlak** (interrupt priority level) Die prioriteit van 'n onderbreking relatief tot ander onderbrekings. Onderbrekingsversoeke met hoër prioriteite word eerste uitgevoer. 66, 68
- onderbrekingsroetine** (interrupt service routine) 'n Roetine of funksie wat uitgevoer word wanneer 'n onderbrekingsversoek plaasvind. 58–62, 64, 66, 68
- onderbrekingsversoek** (interrupt request) 'n Gebeurtenis waarby 'n mikroverwerker versoek word om die uitvoering van die hooflus tydelik te onderbreek, sodat 'n onderbrekingsroetine uitgevoer kan word. 51, 52, 64, 150
- oorgangsband** (transition band) Die oorgang tussen die deurlaatband en die keerband, waar 'n filter verander van 'n hoë na 'n lae aanwins. 41, 42
- oorgangsfrekwensie** (crossover frequency) Die frekwensie waarby die ooplusaanwins van 'n beheerstelsel die 0dB merk kruis. 163, 165
- oormonstering** (oversampling) Die gebruik van meer monsters in 'n steekproefneming as wat nodig is, of 'n hoër monstertempo as wat nodig is. 114, 120
- opgaande flank** (rising edge) 'n Posisie in 'n pulstrein waar die waarde van laag na hoog verander. 96, 97, 102
- partisie** (partition) 'n Gedeelte van 'n stoormedium wat afsonderlik van ander gedeeltes funksioneer. Gewoonlik bevat elke partisie presies een lêergids. 74, 75, 169, 170, 173, 177, 178
- ranggetal** (ordinal) 'n Getal wat aan 'n voorwerp toegeken word om volgorde aan te dui. 67
- rondomtalie** (round robin) Beskrywend van 'n algoritme wat in 'n kring beweeg. 83, 86
- rugsteun** (backup) 'n Duplikaat wat van 'n stelsel of van data gemaak word vir in geval die eerste een faal. 97, 169, 170

- ruilgeheue** (swap memory) 'n Geheue-area wat gewoonlik op groot, stadige media voorkom en gebruik word om virtuele geheue te vergroot wanneer die ewetoeganklike geheue vol is. 87, 93
- ruisverspreiding** (dithering) Die byvoeging van ruis tot 'n sein om die harmoniese vervorming van seine met 'n lae amplitude te verminder. 37, 38
- saamsteltaal** (assembly language) 'n Lae-vlak programmeertaal wat uit instruksies vir 'n spesifieke familie van verwerkers bestaan. Dit word deur 'n samesteller in masjienkode omgeskakel. 52, 68
- selflaaiprogram** (bootloader) 'n Program wat uitgevoer word om 'n bedryfstelsel of mikro-beheerderprogram af te skop. 97
- sirkulêre normaalverdeling** (wrapped normal distribution) Die waarskynlikheidsverdeling wat verkry word wanneer die normaalverdeling om 'n eenheidsirkel gevou word. Dit is die normaalverdeling vir 'n periodiese domein, bv. hoeke. 124
- skeduleringsbeleid** (scheduling policy) Die algoritme waarvolgens 'n bedryfstelsel besluit watter proses volgende toegang tot 'n verwerker sal kry. 83
- skimkrag** (phantom power) 'n Lae-ruis GS-spanning op beide die positiewe en negatiewe lyne van 'n gebalanseerde mikrofoonkabel, wat gebruik word om die kapasitor in 'n kapasitormikrofoon voor te span. 8, 155, 156
- skrip** (script) 'n Program (gewoonlik kort) wat in 'n geïnterpreteerde programmeertaal soos `bash` geskryf is. 'n Lys instruksies wat deur 'n rekenaar uitgevoer moet word. Kort vir 'manuskrip'. 92, 171, 174, 180, 181
- sleggeaard** (ill-conditioned) Beskrywend van 'n matriks wat byna onomkeerbaar is, sodat die berekening van die omgekeerde matriks onderworpe is aan groot numeriese foute. 135, 136, 144
- stoot** (push) Voeg 'n element by 'n waglys. 64–68, 70
- straalskandering** (beamscanning) 'n Ander naam vir “Steered-response power based sound source localisation” 31
- straalvorming** (beamforming) Die samevoeging van seine van verskeie sensors om 'n gerigte sensor te vorm. 31
- stroomhantering** (current rating of the input or throughput) Die maksimum hoeveelheid stroom wat 'n apparaat kan deurlaat of sink terwyl die werksverrigting binne spesifikasies bly. 9, 161
- stroomlewering** (current rating of the output) Die maksimum hoeveelheid stroom wat 'n apparaat kan lewer terwyl die werksverrigting binne spesifikasies bly. 11, 12
- teen-oorvleuelingsfilter** (anti-aliasing filter) 'n Analooaglaagdeurlaatfilter waardeur 'n sein vóór monsterring beweeg, wat oorvleueling in die frekwensiegebied voorkom deur die frekwensieinhoud naby en bo die Nyquist-frekwensie te stop. iii, 36, 41, 42, 149, 152
- uitset** (output) Die data, parameters, seine of strome wat 'n stelsel verlaat. 6, 12–14, 44, 86, 156, 158–163, 165, 166, 179
- uittree** (output) 'n Poort, posisie of kontakpunt waar 'n stelsel verlaat word. 6, 8, 12, 13, 16, 18–20, 22, 37–45, 47, 50, 55, 58, 73, 90, 96, 97, 158, 159, 161, 163, 165, 166

veelsymetring (multilateration) Die kombinerings van meerdere afstandsvlakberamings om 'n posisie te bepaal. 130, 135, 136

venstering (windowing) Die indeling van 'n sein in 'vensters'. Dit beteken dat kort dele van die sein op 'n slag beskou word, asof daar deur 'n venster gekyk word wat oor die definisieversameling van die sein geskuif word. 140

verrevelde (far field) Die gebied ver van 'n sensorskikking, waar die aanname geld dat 'n sein wat uit daardie gebied kom, 'n plat golffront het wanneer dit die sensors bereik. 31, 112, 113, 121, 122, 124

vertrouensgebied (trust region) Die gebied waarbinne 'n model van 'n funksie as geldig beskou word. 117, 118

vervlegte onderbrekings (preemptive interrupts) Onderbrekingshanteringsroetines wat mekaar kan onderbreek. 66

voorvermenigvuldig (premultiply) Matriksvermenigvuldiging van links. 133

vouproduk (convolution) Die tydgebied-resultaat van die vermenigvuldiging van twee seine in die frekwensiegebied, d.w.s. die inverse Fourier-transform van die produk van die Fourier-transforms van twee seine. 115, 126

wedersydse uitsluiting (mutex) 'n Meganisme wat verseker dat die uitvoering van twee stukke kode nie in tyd sal oorvleuel nie. 64, 65, 150

Akronieme

Engelse terme verskyn in hakies.

BTW belasting op toegevoegde waarde 6, 26, 31, 36, 39, 44

CEP foutradius (Circular Error Probability) 100, 102

DFT diskrete Fourier-transform 115, 126

DHCP ‘Dynamic Host Configuration Protocol’ 178

DMA regstreekse geheue toegang (Direct Memory Access) 52, 53, 58, 60–62, 64, 66–69, 76, 95, 150, 152

DT ‘Device Tree’ 18, 73, 88, 94

DTO ‘Device Tree Overlay’ 16, 73–75, 88–90, 93

FFT vinnige Fourier-transform (Fast Fourier Transform) 126, 128

FIFO eerste-in, eerste-uit (First In, First Out) 64, 86

FPGA herprogrammeerbare logikahekskikking (Field-Programmable Gate Array) 51, 151, 152

GND grond 16, 20, 21, 37, 47, 53, 55–57, 73, 92, 100, 102, 151, 158

GPIO veeldoelige in- of uitreepoort (General Purpose Input Output) 16, 18, 51, 55, 56, 58, 61, 68, 69, 72–75, 82, 86, 88–90, 92, 93, 97, 98

GPS globale posisioneringstelsel iii, iv, 3, 4, 7, 19, 55–58, 67, 73–75, 77, 79, 88–91, 93, 96–103, 105, 109, 111, 112, 130, 136, 138–141, 149–151, 153, 180

GPSD GPS-agtergrondproses (GPS daemon) 89–91, 93, 94, 96, 97, 99–101, 150

GPU grafikaverwerker (Graphics Processing Unit) 75

GS gelykstroom 6, 19, 27, 28, 42–44, 46, 155

HDF hoogdeurlaatfilter 5–7

HDMI hoë definisie multimediakoppelvlak (High Definition Multimedia Interface) 75

int24_t 24-bis heelgetal met teken 77, 80

int32_t 32-bis heelgetal met teken 82

IP internetprotokol 89, 91, 92, 178, 179

- LED liguitstralende diode (Light-Emitting Diode) 56–58
- MEMS mikro-elektromeganiese sisteem iii–v, 8, 10, 24, 31, 155
- MOSFET metaaloksiedhalfgeleier veldeffektransistor (Metal-Oxide-Semiconductor Field-Effect Transistor) 161–163, 165
- NCG toegevoegde-gradiënt Newton (Newton Conjugate Gradient) 117, 125–128
- NI beleefdheid (Niceness) 83
- NMEA0183 National Marine Electronics Association: spesifikasie 0183 97, 101, 102
- NP nie-deterministiese polinoomtyd 15
- NTP netwerktidprotokol 89, 92, 102
- NTPD NTP-agtergrondproses (NTP daemon) 75, 89–92, 94, 99, 150
- OSR mate van oor-monstering (Over-Sample Rate) 37, 38
- PGA programmeerbare voorversterker (Programmable Gain Amplifier) 38
- PHD ‘Probability Hypothesis Density’ 144
- PLL fasegesinkroniseerde beheerlus (Phase-Locked Loop) 105, 106, 151
- PPS puls-per-sekonde 55–58, 67, 72–74, 77, 79, 88–91, 96–106, 150, 151
- PR prioriteit 83
- pv Pipe Viewer 87, 88
- RAM ewetoeganklike geheue (Random Access Memory) 69, 70, 72, 74, 87, 88
- RPI Raspberry PI iii, iv, 4, 7–9, 11, 13, 16–20, 51–56, 58, 61–63, 66–70, 72–77, 86–99, 101, 109, 149, 150, 152, 153, 169, 171, 174, 177–181
- RS Reed-Solomon 63
- SD Secure Digital iii, iv, 72, 74, 75, 85, 87, 88, 90, 93, 94, 138, 150, 152, 169–171, 173, 174, 177–179
- SHM gedeelde geheue (SHared Memory) 89, 91, 99
- SMA subminiatur A 97
- SNR sein-na-ruis-verhouding 21, 36, 38, 119, 123, 151
- SPI serie-koppelvlak vir randapparatuur (Serial Peripheral Interface) iii, iv, 20, 36, 38, 51–53, 55, 58–63, 67–70, 73, 74, 76, 77, 80, 82, 85, 86, 88, 93, 95, 99, 150–152
- SPL klankdrukvlak (Sound Pressure Level) 1, 24
- SSH beveiligde tolk (Secure SHell) 74, 92–94, 153, 178–180
- stderr standaardfoutuittree (standard error output) 87
- stdin standaardintree (standard input) 87

- `stdout` standaarduittree (standard output) 82, 87, 171, 174
- STFT korttermyn Fourier-transform (Short Time Fourier Transform) 140
- `sudo` supergebruikersregte 83, 86, 89, 169
- T-NCG vertrouensgebied toegevoegde-gradiënt Newton (Trust-region Newton Conjugate Gradient) 117
- TI Texas Instruments 21, 39, 43–45, 151, 156
- UART universele asinkrone ontvanger en sender (Universal Asynchronous Receiver-Transmitter) 58, 73, 75, 76, 88, 89, 91, 96, 97, 99–102, 150
- `uint16_t` 16-bis heelgetal sonder teken 65, 80, 82
- `uint32_t` 32-bis heelgetal sonder teken 77, 80, 82
- USB universele serie-bus 6, 7, 11, 88, 93, 94, 101, 152, 156
- UTC gekoördineerde universele tyd 101, 103
- UTM Universele Transversale Mercatorprojeksie 140
- WGK wortel van die gemiddelde kwadraat 159
- WiFi draadlose ethernet 4, 5, 92, 94, 149, 152
- WINS Windows Internet Name Service 92
- WS wisselstroom 19, 46, 155

Inhoudsopgawe

Verklaring	ii
Uittreksel	iii
Abstract	iv
Dankbetuiging	v
Woordelys	vi
Akronieme	xi
Inhoudsopgawe	xiv
1 Inleiding	1
1.1 Olifantgeluidnavorsingsgeskiedenis	1
1.2 Motivering	2
1.3 Omvang	3
1.4 Struktuur van die dokument	4
2 Bestaande toerusting	5
2.1 Kommersieel beskikbare stelsels	5
2.2 Motivering vir die ontwerp van 'n eie klankopnemer	6
3 Kragbronne	8
3.1 Intreebeskerming	8
3.2 3A 5V-bron	9
3.3 Lineêre reguleerders	13
3.4 Borduitleg en -vervaardiging	15
3.5 Kragbronbeheerder	16
3.6 Ruis met 'n aktiewe las	18
3.7 Gevolgtrekking	20
4 Mikrofone	23
4.1 Mikrofoonkeuse	24
4.2 Hoogdeurlaatfilter	27
4.3 Stroombaanborde	30
4.4 Windbeskerming	31
4.5 Mikrofoonskikking	31
5 Oudiokoppelvlak	36
5.1 Versyferaar	36
5.2 Voorversterker	39

5.3	Teen-oorvleuelingsfilter	41
5.4	Vervaardiging	46
5.5	Simulasies en metings	47
6	Digitale oudiobuffer	51
6.1	SPI en DMA	51
6.2	PIC16F1847-mikrobeheerder	52
6.3	PIC32MX-mikrobeheerder	53
6.4	Verbindings	53
6.5	SPI-kommunikasie met versyferaar	58
6.6	SPI-kommunikasie met RPI	61
6.7	Draadbestande sirkelbuffers	64
6.8	Onderbrekingsprioriteite	66
6.9	DMA-kanaal-prioriteite	68
6.10	Spanningsmonitor	69
6.11	Toestandsmasjien	69
6.12	Opsomming	70
7	Raspberry Pi	72
7.1	Verbindings	72
7.2	Hardeware-instelling in Linux	73
7.3	UART	75
7.4	SPI	76
7.5	Hoër-vlak funksies vir kommunikasie met die buffer	77
7.6	Hantering van fouttoestande	82
7.7	Program wat monsters haal	82
7.8	Isolasie van 'n verwerkerkern	83
7.9	Skedulering van prosesse	83
7.10	Hantering van in-uit-strome	87
7.11	Geheue-bestuur	87
7.12	SD-kaart	87
7.13	GPS	88
7.14	Tydhou	89
7.15	Skakelaars	90
7.16	SSH en WiFi	92
7.17	Outomatiese opstelling van 'n nuwe RPI	92
7.18	Toekomstige verbeterings	93
8	GPS-module	96
8.1	Keuse van module	96
8.2	Verbindings	97
8.3	Antenna	100
8.4	Konfigurasie	100
8.5	Akkuraatheid	102
8.6	Wiskundige ontleding van sinkronisasiefout	103
8.7	Voorstelle vir verbetering	105
9	Onderstel	107
9.1	Batteryhouer	107
9.2	Houer vir stroombaanborde en skakelaars	107
9.3	Mikrofoonarms	107
9.4	Handvatsel	107

10 Klankbronopsporing	112
10.1 Rigtingberaming met 'n enkele mikrofoonpaar	112
10.2 Triangulasie van rigtingberamings	121
10.3 Hantering van hoekdubbelsinnigheid	122
10.4 Rigtingberaming met 'n nie-lineêre mikrofoonskikking	124
10.5 Nabye-veld-veelsymetring met gesinkroniseerde wyd-gespasieerde mikrofone . .	130
10.6 Toetse met eie klankopnemers	138
11 Gevolgtrekking	149
11.1 Opsommend	149
11.2 Toekomstige werk	150
Bylaes	154
A Die ontwerp van 'n 48V-skimkragbron vir kapasitormikrofone	155
A.1 Spesifikasies	156
A.2 Ontwerp	156
A.3 Toets	166
B Wiskundige bewyse vir spesiale gevalle by veelsymetring	167
B.1 'n Klankbron wat op 'n simmetrie-as lê	167
B.2 'n Klankbron wat op twee simmetrie-asse lê	167
B.3 Die effek van kort mikrofoonafstande	168
C Rugsteunkopieë van SD-kaarte	169
C.1 Beeld die SD-kaart af	169
C.2 Herstel die afbeelding na die SD-kaart	173
D Installeer sagteware op 'n nuwe bedryfstelsel	177
E DVDs	180
Bibliografie	185

Hoofstuk 1

Inleiding

Die uiteindelijke doel van hierdie projek is om die ligging van wilde olifante in hul natuurlike habitat te beraam, sonder enige interaksie met die dier self. Terselfdetyd moet die impak op die habitat geminimeer word.

Ondersoek na bestaande klankopnemers wat vir hierdie doel gebruik kan word, is ingestel. Die ontwerp van 'n eie klankopnemer is gemotiveer en uitgevoer. Dit behels die ontwerp van kragbronne, filters, 'n versyferaarstadium, 'n digitale oudiobuffer en 'n digitale stoormedium. 'n Vergelyking van verskeie mikrofone is gemaak en mikrofone met geskikte lae-frekwensiegedrag is gekies.

1.1 Olifantgeluidnavorsingsgeskiedenis

Berg (1983) het die geluide van olifante in aanhouding ondersoek en 10 verskillende vokaliserings onderskei. Sy het ook bevind dat daar 'n sterk korrelasie is tussen die frekwensie van die geluide en die opgewondenheid van die dier. Payne *et al.* (1986) het die ontdekking gemaak dat baie van hierdie olifantdreungeluide 'n infrasoniese komponent met hoë klankdrukvlakke (SPLs, Sound Pressure Levels) het. Kort daarna het Poole *et al.* (1988) die sosiale konteks waarin hierdie vokaliserings geproduseer word, begin ondersoek en vermoed dat olifante mekaar dalk oor lang afstande kan hoor.

Langbauer *et al.* (1989) het ondersoek ingestel na die reaksies van olifante in aanhouding op olifantgeluide wat deur luidsprekers gespeel word. Daarna het Langbauer *et al.* (1991) die gedrag van wilde olifante in die Etosha Nasionale Park in Namibië probeer beïnvloed deur infrasoniese olifantdreungeluide deur 'n luidspreker te speel. Hulle het bevind dat olifante tot op 2 km op die klanke reageer deur hul ore te lig, te verstar, of die kop heen en weer te beweeg. Olifantbulle het selfs op opnames van maatroepe gereageer deur sover as 'n kilometer reguit in die rigting van die luidspreker te loop.

Garstang *et al.* (1995) en Larom *et al.* (1997b,a) het met behulp van rekenaarmodelle bepaal dat die weerstoestande in die Etosha Nasionale Park, veral snags en in droë tye, ideaal is vir die propagering van lae-frekwensie geluide. Hulle spekuleer dat olifante onder hierdie omstandighede oor 'n afstand van 10 km kan kommunikeer. McComb *et al.* (2003) argumenteer egter dat, alhoewel die olifantdreungeluide dalk oor 10 km nog hoorbaar is, dit nie meer noodwendig die nodige hoë-frekwensie inligting bevat wat nodig is vir nuttige kommunikasie nie. Hulle het wel olifantdreungeluide deur luidsprekers gespeel en vasgestel dat olifantwyfies nie net klanke oor 2,5 km kan herken nie, maar ook kan hoor of dit die roep van 'n familielied is.

Tot dusver fokus die meeste navorsing oor die analise van olifantgeluide op baie spesifieke probleme in beheerde omstandighede: Soltis *et al.* (2005) en Clemens *et al.* (2006) het die uitkenning van individuele olifante op grond van hul vokaliserings ondersoek, maar het gebruik gemaak van mikrofone wat op halsbande van mak olifante in Disney World, Florida, bevestig is. Die eersgenoemde studie het ook bevind dat die emosionele toestand van die olifant uit hul

vokaliserings bepaal kan word. In 'n soortgelyke studie met dieselfde groep olifante het [Leong et al. \(2003\)](#) die olifantgeluide gekarakteriseer en in groepe geklassifiseer. [Stoeger-Horwath et al. \(2007\)](#) het soortgelyke werk met olifantkalfies in 'n dieretuin in Oostenryk gedoen.

Die herkenning van olifantgeluide in natuurlike – en dus raserige – omgewings het eerste in [Venter en Hanekom \(2010\)](#) en [Wijayakulasooriya \(2011\)](#) aandag geniet. [Zeppelzauer et al. \(2013, 2014\)](#) het 'n beter algoritme voorgestel wat nie op spesifieke parameters van die olifantdreungeluide staatmaak nie. Dieselfde metode is deur [Byker \(2015\)](#) geïmplementeer.

Geïntegreerde hardeware vir die opname en analise van olifantgeluide in hul habitat, bestaan egter nie. Daar is ook 'n gebrek aan koste-effektiewe toerusting wat vir sulke navorsing gebruik kan word. Die ontwikkeling van geskikte hardeware sal dus die fokus van hierdie tesis wees.

1.2 Motivering

Die outomatiese akoestiese opsporing van olifante het verskeie toepassings in wildebewaring en die modellering van die gedrag van wilde diere in hul natuurlike habitat. Dit kan ook bydra tot die beveiliging van mense in gebiede waar konflik tussen olifant- en mensbevolkings ontstaan het. Enkele voorbeelde volg:

- [Santiapillai et al. \(2010\)](#) het die oorvleueling van olifanthabitat en landbougrond in Sri Lanka ondersoek. Die steekproefruimte verteenwoordig 65 % van die oppervlakte van Sri Lanka. 65 % van die bevolking in die proef rapporteer dat hul gewasse al deur olifantbulle geplunder is.
- Op Sanparke se webtuise ([Whyte, 2001](#)) word berig dat stropery die sowat 1,3 miljoen olifante wat in 1979 in Afrika gewei het, teen 1989 na 609 000 verminder het. Intussen het stropery verminder, maar die bestuur van olifantbevolkings bly 'n probleem, veral in lande soos Kenia, waar 70 % van wild buite natuurreservate voorkom. Menslike bevolking neem vinnig toe, terwyl voorheen nomadiese mense al hoe meer hul herderlike leefstyl vir landbou verruil. Landbou-aktiwiteite lok wild en veroorsaak konflik tussen mens en dier. Hierdie konflik word as een van die grootste probleme in olifantbewaring beskou. Om die konflik te bekamp, is dit noodsaaklik om die beweging van olifante in hul habitat deurlopend te monitor.
- Dieselfde artikel verduidelik dat olifante, naas die mens, die grootste impak op hul habitat het. Indien hul syfers nie beperk word nie, word die langtermyn oorlewing van verskeie ander spesies bedreig. Hervestiging, beheerde jag of voorbehoeding kan toegepas word om olifantbevolkings te beperk. Ongeag die voordele en nadele wat aan elkeen van hierdie tegnieke verbonde is, is dit duidelik dat diegene wat dit moet toepas, sal baat by die outomatiese opsporing van olifante.
- Volgens [Schoeman \(2017\)](#) poog die Suid-Afrikaanse Nasionale Parke (Sanparke) om olifante se watergebruik te beperk tot spesifieke watergate. Dit word gedoen om homogenisering van die landskap te voorkom, en sodat ander spesies nie by ander watergate verdryf word nie. 'n Outomatiese olifantopspringstelsel sou Sanparke in staat stel om beter besluite te neem oor watter watergate geteiken moet word en te bepaal of hul oplossings effektief is.
- Gesprekke met veeartse en natuurbewaarders het aangedui dat die gedrag van olifante verander wanneer ander diersoorte in hul habitat gestroop word. Dit is moontlik dat die deurlopende monitering van olifantgedrag inligting oor stropingsaktiwiteite sou verkap.

Konvensionele metodes vir die opsporing van diere in hul natuurlike habitat behels gewoonlik dat daar 'n merker of halsband aan die dier vasgemaak word, of dat intensiewe arbeid deur parkbeamptes verrig word:

- **Argos Doppler**¹ is 'n kommersiële satellietstelsel wat seine wat deur die merker aan die dier uitgestuur word, ontvang en met behulp van die Doppler-effek 'n posisieberaming maak. Die beramings word dan op 'n sentrale bediener gestuur. Dit is minder akkuraat as **globale posisioneringstelsel (GPS)** merkers, maar bied die voordeel dat geen verdere kommunikasie met die merker nodig is nadat die posisie beraam is nie.
- Merkers wat met **GPS** werk, verkry die ligging van die dier van satelliete en berg dit vir latere prosessering. Die dier moet weer gevang word om die data af te laai, of die inligting moet via radiokommunikasie na 'n sentrale ontvanger gestuur word. Dieselfde geld vir enige ander oplossing waar data deur 'n apparaat wat aan die dier bevestig is, versamel word.
- Ringe met identifikasienommers kan aan die diere bevestig word. Sodoende kan die langtermyngedrag – soos migrasie – van sekere spesies bestudeer word. Dit werk egter net as die dier later weer gevang word, of as die identifikasienommer met 'n verkyker raakgesien kan word.
- Diere kan te voet of met voertuie agternagesit word, of met behulp van hul mis en spore opgespoor word.
- Kameras kan gebruik word. Indien 'n beweging plaasvind, moet 'n persoon na die videomateriaal kyk om te bepaal watter dier dit is. Andersins kan rekenaarvisie gebruik word om dit outomaties te probeer doen.
- Dierbevolkings kan beraam word deur hul oorblyfsels of ander genetiese materiaal te versamel.

Akoestiese en visuele opsporing bied die voordeel dat individuele diere nie gevang of agternagesit hoef te word nie. Akoestiese metodes werk boonop waar sig beperk is. Aktuele voorbeelde van situasies waar akoestiese stelsels voordele bied bo konvensionele metodes, sluit die volgende in:

- **Elephants Alive (2017)** is 'n organisasie wat hul beywer vir die bewaring van olifante. Hul opsporingspan gebruik halsbande met GPS-toerusting om spesifieke olifante te volg, sodat die gedrag van die olifantroppe bestudeer kan word. Dit is wenslik om interaksie met die olifante te minimeer – iets wat nie moontlik is as daar toerusting aan die dier se liggaam bevestig word nie. Om 'n halsband aan 'n olifant te sit, is boonop 'n tydrowende en duur proses en is traumaties vir die dier. Toerusting word maklik beskadig, en die batterye kan nie maklik vervang word nie.
- Sowat 25–33% van sogenaamde Afrika-olifante is bos-olifante. As gevolg van die digte plantegroei in hierdie habitat, word die aantal olifante gewoonlik na aanleiding van hul mishope beraam. Dit is egter 'n onakkurate proses wat min oor die geslag of ouderdom van die olifante, of die verwantskappe tussen hulle, verklap (Eggert *et al.*, 2003). Akoestiese metodes is hier nuttig omdat sig beperk is.

1.3 Omvang

Die ontwikkeling van 'n akoestiese stelsel bied egter ook 'n paar uitdagings. Hierdie projek poog om die volgende probleme op te los:

- Olifantgeluide het 'n infrasoniese komponent (8–20 Hz) wat nie deur mense gehoor kan word nie. Amper alle mikrofone en klankopnemers filter daardie klanke uit. Daar is spesialistoerusting wat hierdie frekwensiebereik dek, maar dit is ver buite die finansiële bereik van hierdie navorsingsprojek.

¹ Sien <http://www.argos-system.org/>.

- Die stelsel moet battery-aangedrewe wees.
- Baie lae intrinsieke ruis is wenslik. Hoe minder ruis daar is, hoe sensitiewer is die stelsel, sodat olifantgeluide wat van ver af kom en dus baie sag is, beter bespeur kan word.
- Die stelsel moet draadloos kan kommunikeer, byvoorbeeld deur **draadlose ethernet (WiFi)**. As die apparaat in wildgebied geïnstalleer word, moet data veilig afgelaai kan word sonder om 'n voertuig te verlaat.
- Die stelsel moet baie akkuraat tydhou, sodat klankmonsters van een installasie met dié van 'n ander gekorreleer kan word.
- Goedkoop komponente word verkies bo duur, gekalibreerde klanktoerusting, omdat die projek toeganklik moet wees vir toekomstige studente om aan te werk. Daar bestaan ook 'n reële gevaar dat 'n olifant tydens die toetsfase een van die stelsels sal beetkry.

Aan die ander kant is daar ook probleme wat buite die omvang van hierdie projek is:

- Beskerming van die stelsel teen vernietiging deur olifante of ander wilde diere.
- Kamoefflering, sodat diere nie bewus is van die stelsel nie.
- Toetse om te kyk of die teenwoordigheid van so 'n apparaat 'n negatiewe effek op die gedrag van diere het.
- Vermindering van eggo's en weergalming – Daar word aangeneem dat eggo's en weergalming nie 'n groot rol in die bosveld sal speel nie. In beboste gebiede mag dit 'n groter rol speel. Hierdie aanname kan eers getoets word nadat 'n werkende prototipe van hierdie projek gebou is.

1.4 Struktuur van die dokument

Verskeie bestaande klankopnemers word in §2 geassesseer. Die ontwerp van kragbronne word in §3 bespreek. Mikrofone word gekies en vergelyk in §4. §5 bespreek die ontwerp van 'n oudiokoppelvlak wat versterkers, filters en versyferaars insluit. Die ontwerp en programmering van 'n buffer om die data van die versyferaar afkomstig, te hanteer en berg, word in §6 bespreek. Die opstelling en programmering van 'n **Raspberry PI (RPI)** om die versamelde data te prosesseer en berg, word in §7 bespreek. In §8 word die opstelling van 'n **GPS**-module vir die sinkronisasie van monsterring oor lang afstande bespreek. Die vervaardiging van 'n onderstel om al die verskillende komponente te huisves, word in §9 beskryf. Algoritmes vir die opsporing van olifantdreungeluide met hierdie stelsel word in §10 bespreek. Data word versamel en simulaties en toetse word uitgevoer. §11 maak gevolgtrekkings oor die projek en beskryf toekomstige werk wat tot hierdie projek kan bydra.

Hoofstuk 2

Bestaande toerusting

Die doel van die projek, soos in §1 beskryf, is om posisie-inligting van olifante te verkry. Dit bring die volgende vereistes vir die klankopnemer mee:

- Meerdere gesinkroniseerde mikrofoonkanale is nodig om die aankomshoek van klanke te bepaal.
- 'n Betreklik plat frekwensieweergawe vir die volle omvang van tipiese olifantdreungeluide (8 Hz–500 Hz) word verlang.
- Die groottegeweergawe en fasegeweergawe moet konsekwent wees tussen kanale.
- Om klank oor lang afstande te bespeur, word 'n lae intrinsieke ruis verlang.
- Die stelsel moet volledig draadloos wees. Dit vereis *a)* 'n battery-aangedrewe stelsel *b)* met 'n draadlose verbinding soos **WiFi**.
- Die stelsel moet posisie- en tydbewus wees om klankopnames met ander opnemers akkuraat te kan sinkroniseer.

2.1 Kommersieel beskikbare stelsels

Om 'n herontwerp van die wiel te probeer vermy, is ondersoek na kommersieel beskikbare stelsels ingestel. Elke opnemer word aan die bostaande vereistes gemeet. Tabel 2.1 toon 'n lys van opnemers en die spesifikasies waaraan elkeen voldoen, al dan nie. Die kolomme bevat die volgende inligting:

- Aantal **mikrofoonvoorversterkers**.
- Maksimum aantal verskillende **intrees** wat sonder vermenging opgeneem kan word.
- Gespesifiseerder **frekwensiegedrag** volgens die vervaardiger.
- Prys volgens www.sweetwater.com in **USD**.
- Plaaslike prys, waar plaaslike verskaffers gevind is, in **ZAR**.

'n Opnemer hoef nie aan al die vereistes te voldoen om vir hierdie projek bruikbaar te wees nie. 'n Bestaande opnemer kan byvoorbeeld maklik aangepas word om battery-aangedrewe te wees, as dit normaalweg van 'n 220 volt-kragbron gebruik maak. Ander eienskappe, soos frekwensiegedrag en die aantal kanale, kan moeiliker verander word.

Sommige spesifikasies is onbekend en kan eers gemeet word indien die toerusting aangeskaf word. Die meeste vervaardigers bied geen spesifikasie van frekwensiegedrag buite die 20 Hz–20 kHz omvang nie. Soms is daar 'n opsie om die 'low cut'¹ af te skakel. Dit is te wagte dat die meeste

¹ 'n **HDF**, tipies met 'n afsnyfrekwensie tussen 50 Hz en 100 Hz.

Produk	mik	in	Frekwensiegedrag	USD	ZAR
Allen & Heath ZEDi-10	4	4	30,0 Hz–10,0 kHz; +0,5 dB; –1,0 dB	200	3840
Behringer FCA1616	4	8	40,0 Hz–10,0 kHz; +0,0 dB; –3,0 dB	250	4080
Behringer U-PHORIA UMC1820	8	8	30,0 Hz–10,0 kHz; +0,5 dB; –1,0 dB	300	
Boss BR800	4	4	20,0 Hz–20,0 kHz; +1,0 dB; –3,0 dB	450	
Cymatic LR-16	0	16	20,0 Hz–20,0 kHz; $\pm 0,2$ dB	500	
iConnectivity iConnectAUDIO4+	4	4	20,0 Hz–20,0 kHz;	300	
PreSonus AudioBox 1818VSL	8	8	22,0 Hz–20,0 kHz; $\pm 0,25$ dB	400	7300
PreSonus AudioBox 44VSL	4	4	22,0 Hz–20,0 kHz; $\pm 0,25$ dB	250	4750
Tascam DP-24SD	8	8	20,0 Hz–20,0 kHz; $\pm 1,0$ dB	420	
Zoom F4	4	6	80,0 Hz–10,0 kHz; +0,5 dB; –1,5 dB	650	

Tabel 2.1: Kommersieel beskikbare multikanaal klankopnemers waarvoor die frekwensiegedrag gespesifiseer word. Al die apparate gebruik **USB** om data oor te dra. Die kolomme **mik** en **in** dui onderskeidelik die aantal mikrofoonvoorversterkers en die aantal kanale wat afsonderlik gemonster kan word aan. Pryse word in **USD** en **ZAR** (**BTW** ingesluit) gegee waar moontlik.

opnemers in elk geval nog 'n **hoogdeurlaatfilter (HDF)** sal hê om **gelykstroom (GS)** foute wat deur die interne versterkers veroorsaak word, uit te haal. Die afsnyfrekwensie van hierdie filters is egter oor die algemeen onbekend. Plat frekwensiegedrag laer as 20 Hz kan dus in geen van die ondersoekte gevalle gewaarborg word nie.

2.2 Motivering vir die ontwerp van 'n eie klankopnemer

Om 'n goeie besluit te kan maak oor watter toerusting aangeskaf moet word, is verskeie stukke beskikbare klanktoerusting eers getoets om 'n aanduiding te kry van die tipiese laagafsnijfrekwensie van klanktoerusting wat vir menslike gehooromvang (tipies kleiner as 20 Hz–20 kHz) ontwerp is:

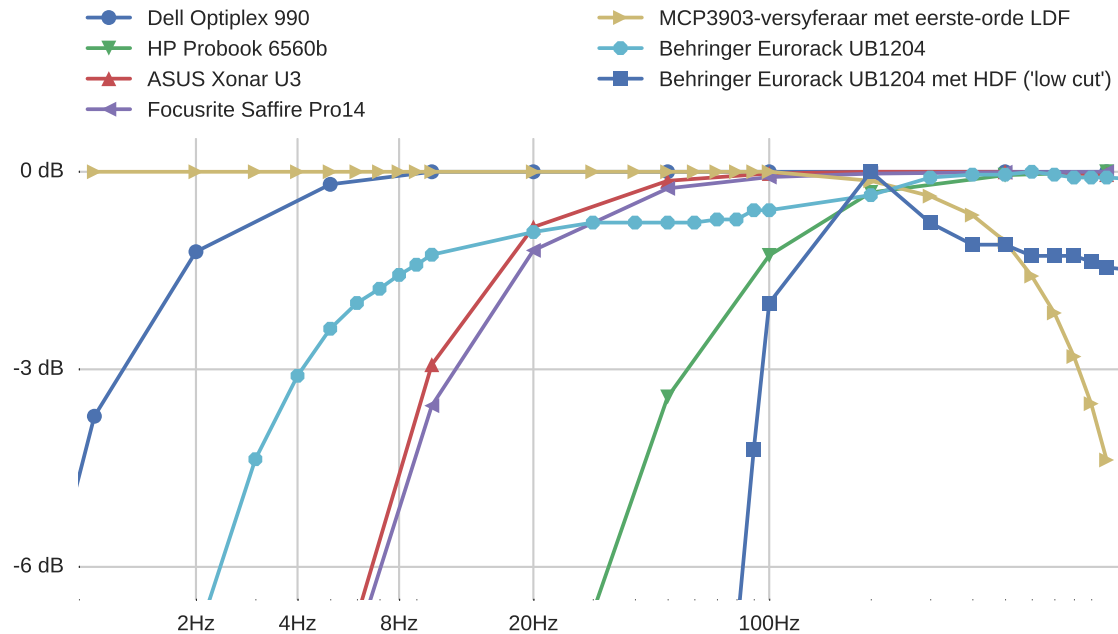
- Voorste mikrofoon**intree** van 'n Dell Optiplex 990-rekenaar.
- Mikrofoon**intree** van 'n HP Probook 6560b-skootrekenaar.
- Mikrofoon**intree** van 'n eksterne ASUS Xonar U3 **USB**-klankkaart.
- Voorversterker-**intree** van 'n Focusrite Saffire Pro14-oudiokoppelvlak.
- Kanaal 1 van 'n Behringer Eurorack UB1204-klankversterker, met én sonder die **HDF**.

Al die toerusting is getoets deur 'n seingenerator aan die **intree** te koppel, sinusvormige seine van 1 Hz tot 1 kHz op te wek en vir elke frekwensie die amplitude by die **uittree** te meet. Die Dell-, HP- en ASUS- en Focusrite-apparate is getoets deur die **intreesin** met Audacity² op te neem. Die **uitset**sin van die Behringer-klankmenger is gemeet deur die 'Main L'-**uittree** aan 'n ossiloskoop te koppel. Die resultate word in Figuur 2.1 getoon.

Die Dell Optiplex 990 het 'n verbasend goeie lae-frekwensiegedrag, met 'n –3 dB-afsnijfrekwensie onder 2 Hz. Dit is natuurlik onprakties om 'n rekenaar vir die finale projek te gebruik, maar dit sou moontlik wees om die stroombaan van dié rekenaar se klankkaart te ontleed en na te boots.

Die klankmenger se frekwensieweergawe, met die **HDF** afgeskakel, het ook 'n geskikte frekwensiegedrag en kan dus direk aan 'n versyferaar gekoppel word. Hierdie oplossing is egter ook onprakties, omdat die stelsel battery-aangedrewe moet wees.

² Audacity is oopbron oudioverwerkingsagteware. Beskikbaar by <http://www.audacityteam.org/>.



Figuur 2.1: Die frekwensieweergawe van verskeie stukke klanktoerusting wat beskikbaar was. Hierdie appaarte is nie geskik vir gebruik in hierdie projek nie, maar gee 'n aanduiding van die tipiese laagafsnijfrekwensie van klanktoerusting wat vir die menslike gehoor ontwerp is. Die frekwensieweergawe van 'n MCP3903 versyferaar sonder 'n HDF is ter wille van vergelyking ingesluit.

Die ASUS Xonar U3 is 'n klein apparaat wat by 'n RPI se USB-poort ingeplug kan word en sou dus 'n praktiese oplossing kon bied. Siende dat die fundamentele frekwensie van olifantdreungeluide so laag soos 8 Hz kan wees, is die $-4,5$ dB-weergawe by daardie frekwensie egter nie ideaal nie. Dit het ook slegs twee intreekanale, wat beteken dat verskeie eenhede in parallel gekoppel moet word om 'n mikrofoonskikking te kan opneem. Omdat die ASUS deur USB werk, is daar geen waarborg dat die monsters van twee sulke appaarte gelyktydig geneem word nie, selfs al is beide aan dieselfde rekenaar gekoppel.

Dit is belangrik om op te let dat geen van die beskikbare klankopnemers in Tabel 2.1 of Figuur 2.1 'n sinkronisasie-funksie het waarmee opnames tussen verskeie opnemer-eenhede akkuraat gesinkroniseer kan word nie. Mikrofone wat ver uitmekaar is kan nie aan dieselfde multikanaal klankopnemer gekoppel word nie. Die monsters moet dus eers gesinkroniseer word voordat die opnames gebruik kan word om die ligging van 'n klankbron te bepaal. Siende dat olifante tot 80 km per dag kan beweeg, is die langafstandsinkronisasie van mikrofone onontbeerlik.

Die bogenoemde probleme het gelei tot die besluit om 'n eie klankopnemer te bou wat vir hierdie projek pasgemaak is. 'n Basiese stroombaanbordjie met 'n MCP3903-versyferaar en 'n eerste orde passiewe laagdeurlaatfilter (om oorvleueling te voorkom) is gebou om die konsep te toets. Figuur 2.1 sluit die intreefrekwensieweergawe van hierdie stroombaan ook in. Die frekwensiegedrag is ideaal tot en met 0 Hz. Daar sal gepoog word om hierdie versyferaar met 'n GPS-module te integreer, wat die akkurate tydtempels vir sinkronisasie sal verskaf.

Die gedetailleerde ontwerp van 'n eie klankopnemer word in §3–§9 beskryf.

Hoofstuk 3

Kragbronne

Om in die veld te kan werk, moet die stelsel battery-aangedrewe wees. 'n Goedkoop en beskikbare opsie is 12 V-loodsuurbatterye. In teenstelling met litium-batterye is loodsuurbatterye eenvoudig om te laai: 'n 13 V–14 V spanningsbron kan in parallel met die battery geskakel word. Die ontwerp van 'n batterylaai is buite die strekking van hierdie tesis.

Die **RPI** benodig 'n 5 V-bron en reguleer self daardie spanning na 3,3 V. Die 3,3 V lyn kan tot 500 mA lewer, wat genoeg is om 'n mikroverwerker, versyferaar, filters, voorversterkers en mikrofone aan te dryf.

Sommige mikrofone benodig ook **skimkrag** tot en met 48 V. Ander mikrofone, soos **mikro-elektromeganiese sisteem (MEMS)** mikrofone, benodig weer 'n lae-spanning (gewoonlik 1,8 V), lae-ruis toevoer op 'n aparte lyn as die sein.

Aanvanklik sou kapasitormikrofone gebruik word, en 'n kragbron met twee kapperbane is gebou, waar een **uittree** 2,6 A teen 5 V en die ander 160 mA teen 48 V lewer. Dit is 'weergawe A' en word in Figuur 3.1 beskryf.

Kapperbane veroorsaak ruis, en aktiewe laste soos die **RPI** se verwerker (§7) induseer ruis op die toevoerspanningslyne. Hierdie ruiserige 5 V- en 3,3 V-lyne word gebruik om die versyferaar aan te dryf. Dit voeg ruis by die versyferde seine en veroorsaak versyferingsfoute.

Aanvanklik is 'n **ChipKIT uC32** vir ontwikkeling gebruik, maar later is daar besluit om 'n eie mikrobeheerderbaan te ontwikkel (sien §6). Daar is ook besluit om eerder **MEMS**-mikrofone as kapasitormikrofone te gebruik (sien §4). In 'n poging om die ruis in die toevoerspanning van die versyferaar te verminder, is die kragbron herontwerp: Daar is weggedoen met die 48 V-bron, en 3 lineêre spanningsreguleerders is bygevoeg:

- a) 50 mA 5 V vir die filters, voorversterkers en die versyferaar se analoogtoevoerspanning.
- b) 50 mA 3,3 V vir die versyferaar se digitale toevoerspanning.
- c) 50 mA 1,8 V vir die **MEMS**-mikrofone.

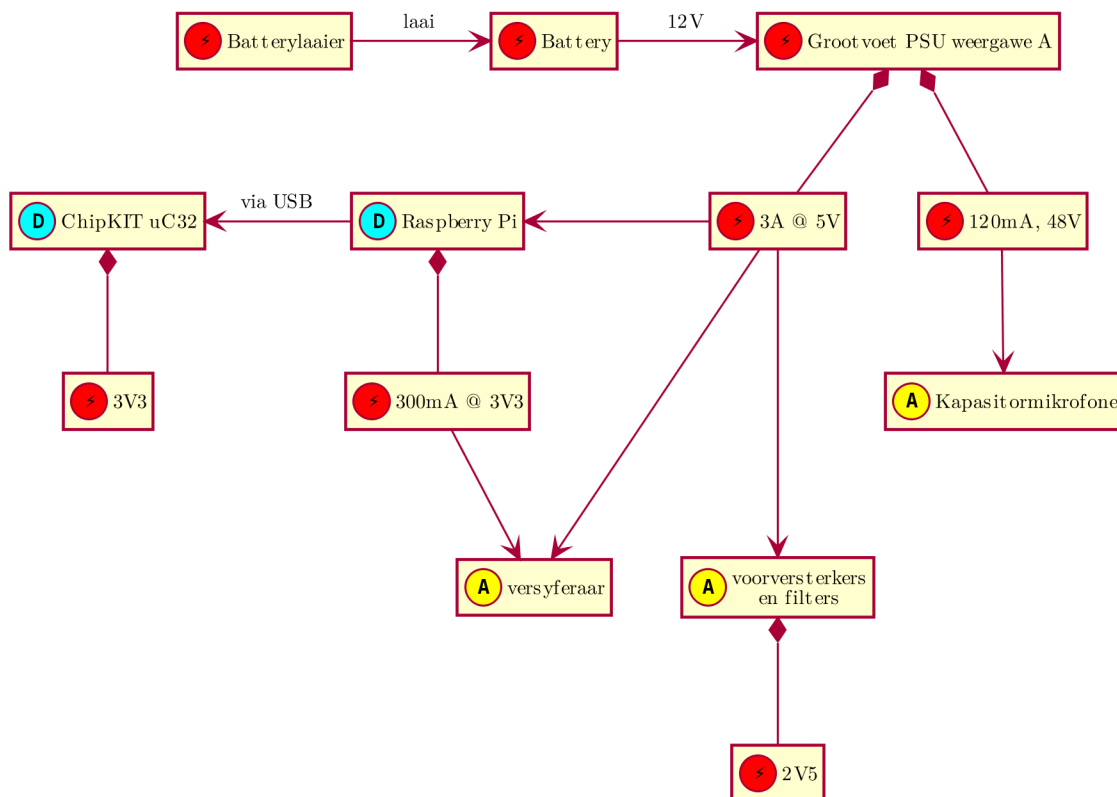
Dit heet 'weergawe B' van die kragbron en word in Figuur 3.2 beskryf.

Die volgende afdelings beskryf die gedetailleerde ontwerp van weergawe B van die kragbron. Omdat die 48 V-bron nie deel vorm van die finale projek nie, is die ontwerp daarvan uitgeskuif na Bylae A. Die ruis in die stelsel word in §3.6 ondersoek.

3.1 Intreebeskerming

Die stroombaan moet teen verkeerde **intreespannings** beskerm word. Die mees algemene foute is om 'n te hoë **intreespanning** aan te lê, of om die **intreespanning** verkeerd te koppel.

'n Serie-smeltdraadje (oftewel sekering) met 'n dubbele 15 V Zener-diode word as 'n eenvoudige beskerming gebruik. Die stroombaandiagram word in Figuur 3.3 getoon.



Figuur 3.1: UML-klasdiagram vir weergawe A van die kragbron. Hier word die geskakelde 5 V en 3,3 V kragbronne gebruik om beide die digitale komponente (RPI en mikrobeheerder) en analoogkomponente (mikrofone, voorversterkers, filters en versyfaars) aan te dryf. Die ruis op hierdie kragbronne word egter nie voldoende deur die analoogkomponente verwerp nie en verhoog dus die ruis in die versyferde sein. Dit het aanleiding gegee tot die ontwerp van weergawe B in Figuur 3.2.

Die gekose diode is 'n Panasonic DZ4J150K0R. Dit is twee 15 V Zener diodes in parallel in een pakkie. Elke diode het 'n toleransie van 5 % en 'n volgehoue stroomhantering van 0.2W.

'n Simulasie met behulp van LTSpice (Figuur 3.4) wys dat 'n intreespanning van 15,6 V die sekering sal laat smelt, selfs as daar geen lasstroom is nie.

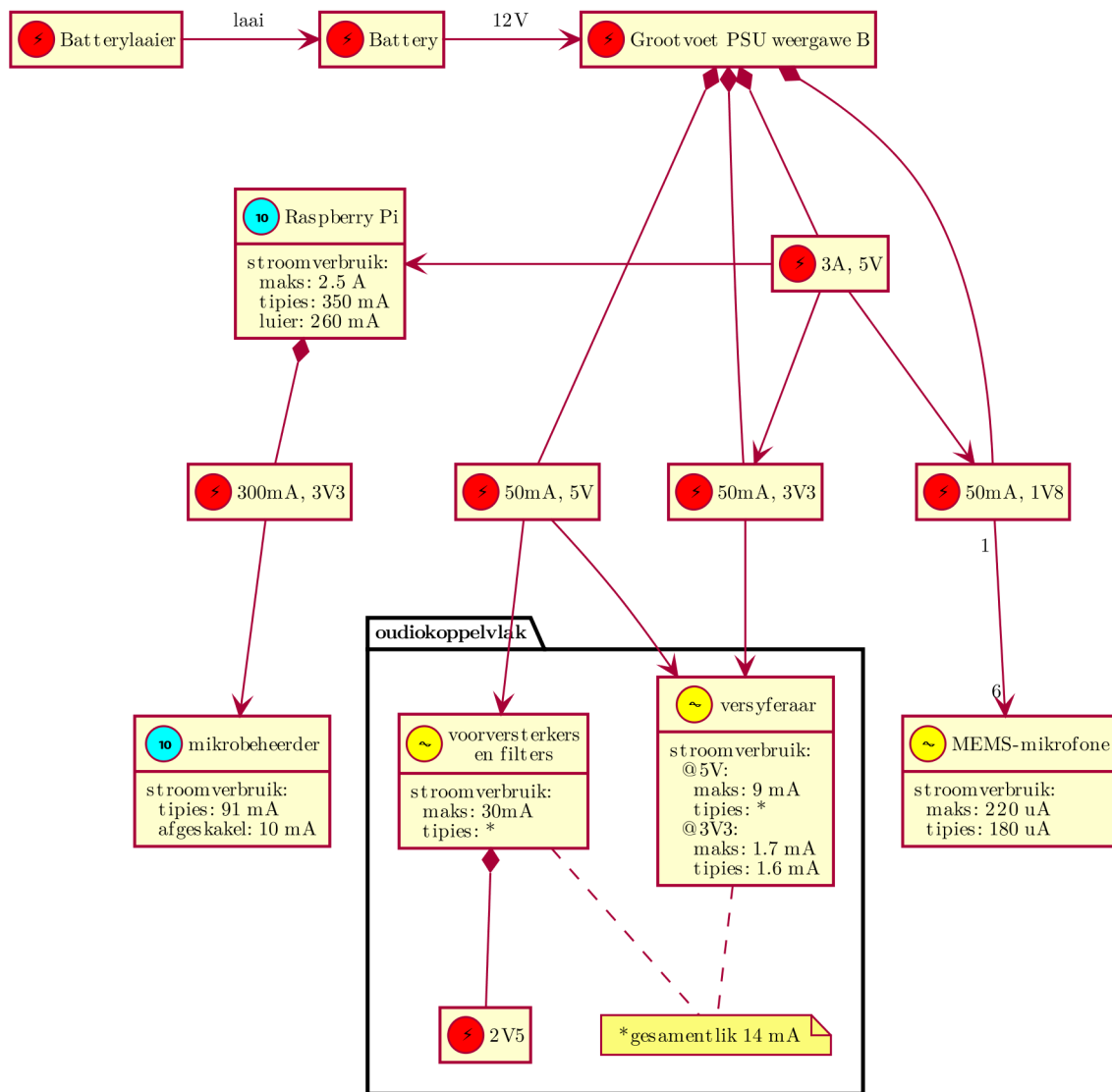
'n Moontlike probleem met hierdie beskermingsmeganisme is dat die Zener-diodes sal uitbrand indien die draadjie nie gou genoeg smelt nie. In die ergste geval word die uitgebrande Zener-diodes 'n oopsluiting eerder as 'n kortsluiting, en dan is daar glad nie meer spanningsbeskerming nie. Vir 'n kommersiële produk sal 'n meer gevorderde beskermingsmeganisme dus gebruik moet word. Vir hierdie prototipe is dit egter belangrik om die aantal komponente laag te hou. Dit is maklik om die smeltdraadjie te vervang indien 'n fout tog plaasvind. Loodsuurbatterye kom ook niks oor as dit vir 'n kort tydjie baie stroom moet lewer terwyl die draad smelt nie.

3.2 3A 5V-bron

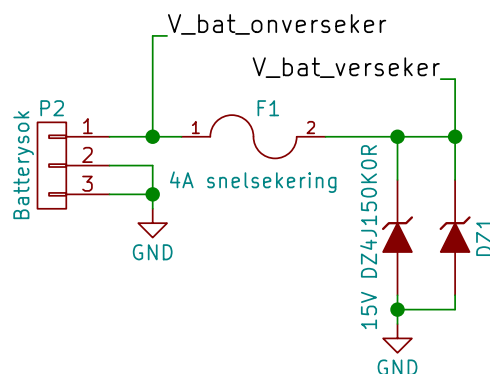
3.2.1 Spesifikasies

'n Loodsuurbatterie met 'n nominale spanning van 12 V kan by tye tot 15 V oor die terminale hê terwyl dit gelaai word. In 'n motorvoertuig kan die dinamo baie hoër spanningspieke veroorsaak, maar ons apparaat sal waarskynlik eerder vanaf 'n laboratoriumkragbron of met sonselle gelaai word.

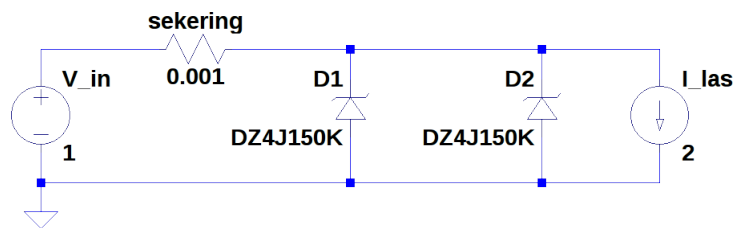
Tabel 3.2.1 dui die stroomverbruik van verskillende RPI-modelle aan. Die kolomme dui die volgende aan:



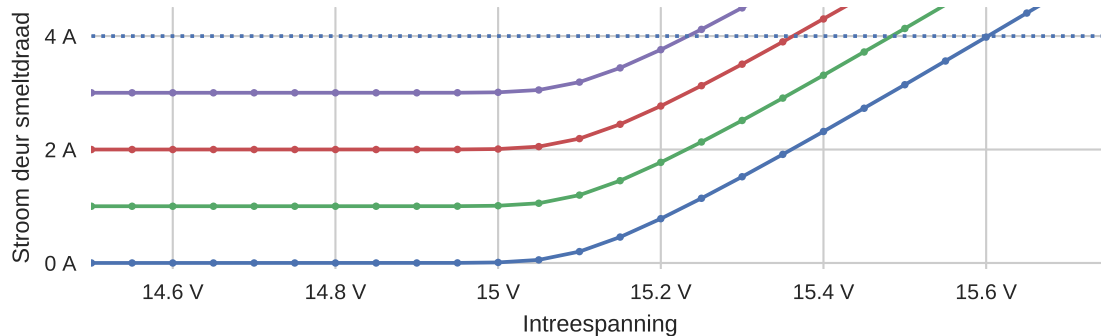
Figuur 3.2: UML-klasdiagram vir weergawe B van die kragbron. Hierdie ontwerp verbeter op weergawe A in Figuur 3.1 deur drie lineêre reguleerders by te voeg om lae-ruis 3,3 V en 5 V kragbronne vir die analoogkomponente te verskaf. Die 48 V kragbron vir die kapasitormikrofone is vervuil vir 'n 1,8 V lineêre reguleerder, omdat die kapasitormikrofone later vir **MEMS**-mikrofone vervuil is. Met behulp van 'n verskuifbare oorbrugging kan die 1,8 V en 3,3 V kragbronne óf direk vanaf die 12 V battery gereguleer word vir laer ruis, óf vanaf die 5 V kapperkragbron vir hoër effektiwiteit.



Figuur 3.3: Die intreebeskerming vir die kragbron maak van 'n 4 A snelsekering en 'n 15 V Zener-diode gebruik. Indien die intreespanning die Zener-spanning oorskry of negatief is, behoort daar genoeg stroom te vloei om die sekering te laat smelt.



(a) Stroombaan vir die simulاسie.



(b) Resultaat van die simulاسie. Elke lyn stel 'n spesifieke waarde van I_{las} voor. Die punte op elke lyn is die hoeveelheid stroom deur die sekering by die gegewe V_{in} . Die diodes begin gelei wanneer die intreespanning hoër as 15 V word. Die sekering behoort te smelt sodra daar meer as 4 A vloei.

Figuur 3.4: Intreebeskerming, met LTSpice gesimuleer. 'n Weglaatbaar klein weerstand is gebruik om die stroom deur die sekering te meet. Die simulاسie-opdrag is `.dc V_in -18 18 0.05 I_las 0 3 1`.

RPI-model	Aanbevole	Randapparatuur	Tipies
A	700 mA	500 mA	200 mA
B	1,2 A	500 mA	500 mA
A+	700 mA	500 mA	180 mA
B+	1,8 A	600 mA of 1,2 A (verstelbaar)	330 mA
2 B	1,8 A	600 mA of 1,2 A (verstelbaar)	
3 B	2,5 A	1,2 A	400 mA

Tabel 3.1: Stroomvereistes van verskeie RPI-modelle. Bron: <https://www.raspberrypi.org/help/faqs/#powerReqs>. Die aanbevole minimum stroomleweringvermoë van die kragbron word in kolom **aanbevole** aangedui. Die **randapparatuur**-kolom dui die hoeveelheid stroom aan wat beskikbaar is vir apparate wat by die RPI se USB-poorte ingeprop word. Die **tipiese** stroomverbruik van die RPI self word in die laaste kolom aangedui.

- **Aanbevole stroomlewering** van 5 V-bron
- Maksimum stroomverbruik van **USB-randapparatuur**
- **Tipiese** stroomverbruik sonder toebehore

Om enige RPI saam met randapparatuur te kan aandryf, is 'n stroomlewering van 3 A wenslik.

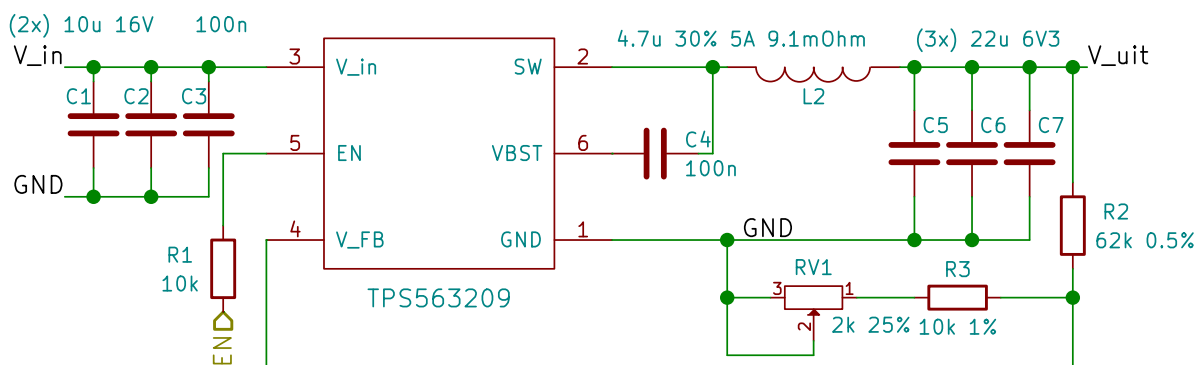
3.2.2 Ontwerp

In 'n ondersoek na gepaste kapper-reguleerders het die TPS563209DDCT van Texas Instruments uitgestaan (Texas Instruments, 2014). Tabel 3.2 som die belangrikste spesifikasies van hierdie komponent op.

Die stroombaan is aan die hand van §9 van die datablad, “Toepassing en implementasie”, ontwerp. Figuur 3.5 toon die finale stroombaandiagram.

Simbool	Beskrywing	Min.	Tipies	Maks.	Eenheid
V_{in}	Insetspanning	4,5	–	17	V
V_{uit}	Uitsetspanning	0,76	–	7	V
I_{uit}	Stroomlewering	–	–	3	A
f_{SW}	Skakelfrekwensie	–	650	–	kHz
V_{FB}	Drumpelspanning by terugvoerpen	758	765	772	mV
I_{FB}	Stroom gesink deur terugvoerpen	–	0	± 100	μA

Tabel 3.2: Noemenswaardige spesifikasies van TPS563209DDCT, uit die datablad geneem (Texas Instruments, 2014).



Figuur 3.5: Die 3 A 5 V-kapperkragbron.

Die datablad beveel aan dat $V_{\text{in}} > \frac{V_{\text{uit}}}{0,65}$, sodat die **dienssiklus** minder as 65% bly. Vir die gewenste **uitset**spanning van 5 V moet die toevoer dus ten minste 7.69 V wees.

Die uittreespanning word met behulp van die spanningsdeler in Vergelyking 3.1a bepaal. Vergelyking 3.1b gee die drywing wat in hierdie terugvoernetwerk verkwis word, onder die aanname dat die stroom wat by die FB-pen invloed weglaatbaar klein is.

$$V_{\text{uit}} = V_{\text{FB}} \left(1 + \frac{R_2}{R_3} \right) \quad (3.1a)$$

$$P_{\text{FB}} = \frac{V_{\text{uit}}^2}{R_2 + R_3} \quad (3.1b)$$

Hoër weerstande veroorsaak dus hoër effektiwiteit, maar laer weerstande maak V_{uit} meer immuun teen veranderinge in I_{FB} . Weerstand R_3 word geïmplementeer as 'n gewone weerstand in serie met 'n verstelbare weerstand, sodat die **uittree** noukeurig gestel kan word om presies 5 V te wees, selfs al is die weerstande nie akkuraat nie en $I_{\text{FB}} \neq 0$.

Komponentwaardes vir die LC-filter, naamlik L_1 , C_5 , C_6 en C_7 word volgens die aanbevelings in die datablad gekies.

Die **intree**kapasitor moet volgens die datablad se aanbevelings 'n keramiekkapasitor groter as $10\mu F$ wees. Keramiekkapsitors van daardie grootte is moeilik om te vind of duur, en daarom word twee (C_1 en C_2) in parallel gebruik. 'n Derde, kleiner keramiekkapasitor (C_3) word in parallel bygevoeg om hoë-frekwensie toevoer-lynregulasie te verbeter. Kapasitor C_4 word eenvoudig deur die datablad voorgeskryf.

Die finale keuse van komponente word in Tabel 3.3 gegee, en die uiteindelijke spesifikasies van die kragbronn verskyn in Tabel 3.4.

Die ontwerp van die borduitleg en die vervaardiging van die stroombaan word in §3.4 bespreek.

Komponent	Waarde	
R_2	$10\text{ k}\Omega \pm 0,5\%$	
R_3	$62\text{ k}\Omega \pm 1\%$	
$R_{3,\text{pot}}$	$2\text{ k}\Omega \pm 25\%$	
L_1	$4,7\text{ }\mu\text{H} \pm 30\%$	ESR = $9,1\text{ m}\Omega$
C_1, C_2	$10\text{ }\mu\text{F} \pm 10\%$	X5R keramiekkapasitors, 16 V
C_3, C_4	$100\text{ nF} \pm 20\%$	keramiekkapasitor
C_5, C_6, C_7	$22\text{ }\mu\text{F} \pm 20\%$	X5R keramiekkapasitors, 6,3 V

Tabel 3.3: Komponentwaardes vir die 5 V-kapperkragbron in Figuur 3.5.

Simbool	Beskrywing	Min.	Tipies	Maks.	Eenheid
P_{FB}	Drywingsverkwisting in terugvoernetwerk	–	382,5	–	μW
$V_{\text{uit,min}}$	Minimum stelbare uitset spanning	4,51	–	4,95	V
$V_{\text{uit,maks}}$	Maksimum stelbare uitset spanning	5,44	–	5,58	V
V_{in}	Aanvaarbare toevoerspanning	7,69	12	17	V
$I_{\text{L,pp}}$	Rimpelspanning deur induktor	–	1,09	–	A
$I_{\text{L,piek}}$	Piekstroom deur induktor	–	3,55	–	A
$I_{\text{L,WGK}}$	WGK-stroom deur induktor	–	3,02	–	A

Tabel 3.4: Uiteindelijke teoretiese spesifikasies van die 5 V-kapperkragbron in Figuur 3.5.

3.2.3 Toets

'n Reostaat is aan die 5 V-**uittree** van die kragbron gekoppel, terwyl die 48 V-bron, wat op dieselfde bord gesoldeer is, geen las gehad het nie. Die weerstand is inkrementeel met die hand vermeerder, en 'n gemiddelde **uittreespanning** is met 'n ossiloskoop gemeet. Die resultate in Figuur 3.6 toon dat die gemiddelde **uitsetspanning** 5 V is. Die **uitsetspanning** bly binne 0,1 V van die spesifikasie.

Terselfdetyd is die **insetstroom** gemeet, sodat die algehele effektiwiteit van die kragbron bereken kan word. Die effektiwiteit bereik 'n maksimum van 94 % by 1–2 A, wat die tipiese stroomverbruik van 'n RPI is.

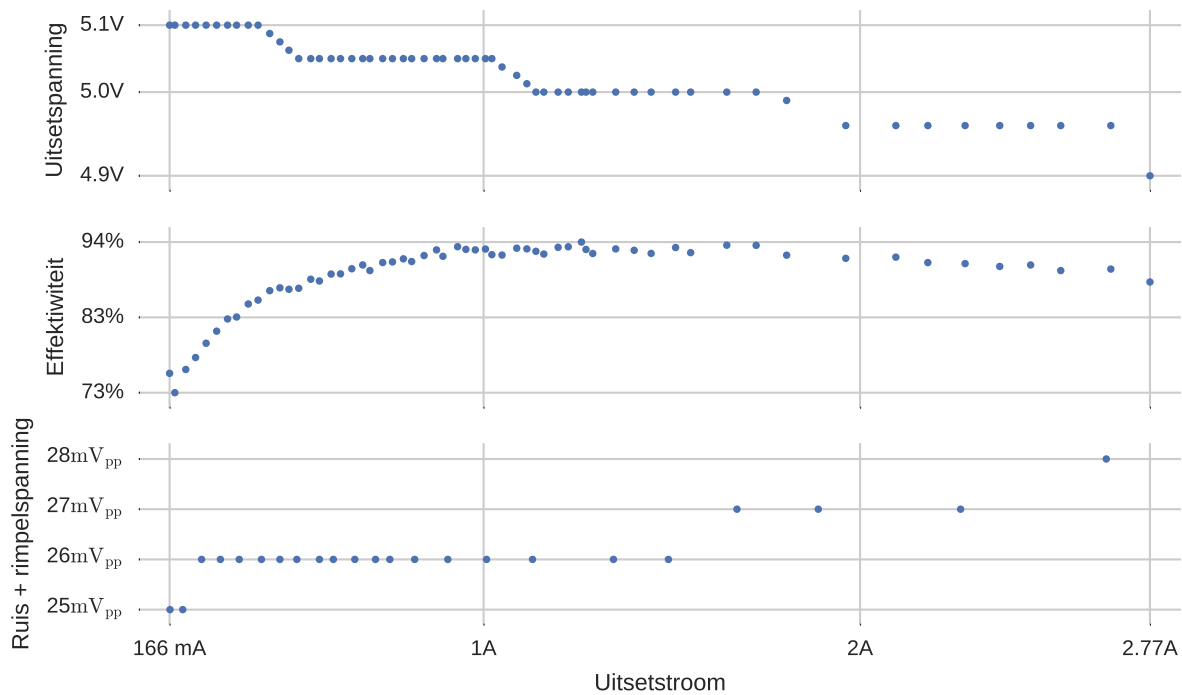
Met dieselfde opstelling as hierbo, is die piek-tot-piek ruis plus rimpelspanning by die **uittree** gemeet. Die ruis en rimpelspanning neem effens toe, saam met die **uitsetstroom**, tot by 'n maksimum van 28 mV_{pp} . Die ruis is egter heelwat erger met 'n aktiewe las soos 'n **RPI** of 'n mikrobeheerder. Dit word verder in §3.6 bespreek.

Daarna is die reostaat gestel om 2 A teen 5 V uit die bron te trek, terwyl die **insetspanning** varieer word. Figuur 3.7 toon die resultate. Soos verwag bereik die **uitsetspanning** eers 5 V met 'n **insetspanning** van 7,5 V of meer. Die ruis by die **uittree** word ook meer wanneer die **intreespanning** te laag daal. Met 'n **insetspanning** laer as 4 V begin die induktor hoorbaar vibreer, maar die stroombaan word nie beskadig nie.

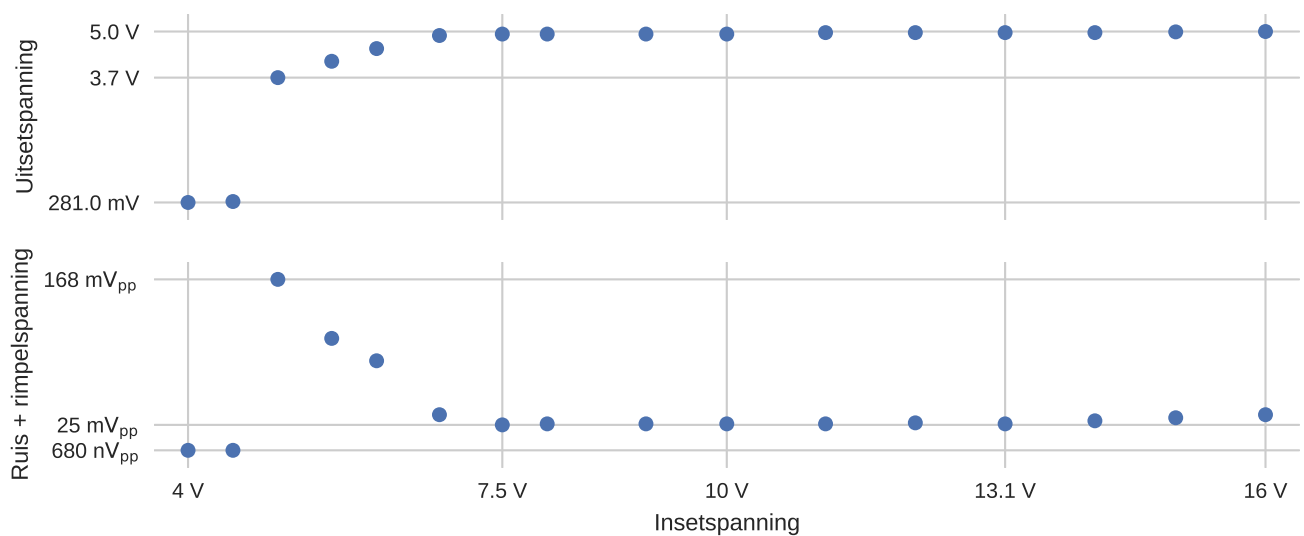
Hierdie kragbron is dus stabiel en veilig. Dit beskerm egter nie die battery teen oormatige dreinerings nie. Om hierdie probleem op te los, word die 5 V-**uittree** gemonitor. Die werking hiervan word in §6.10 beskryf.

3.3 Lineêre reguleerders

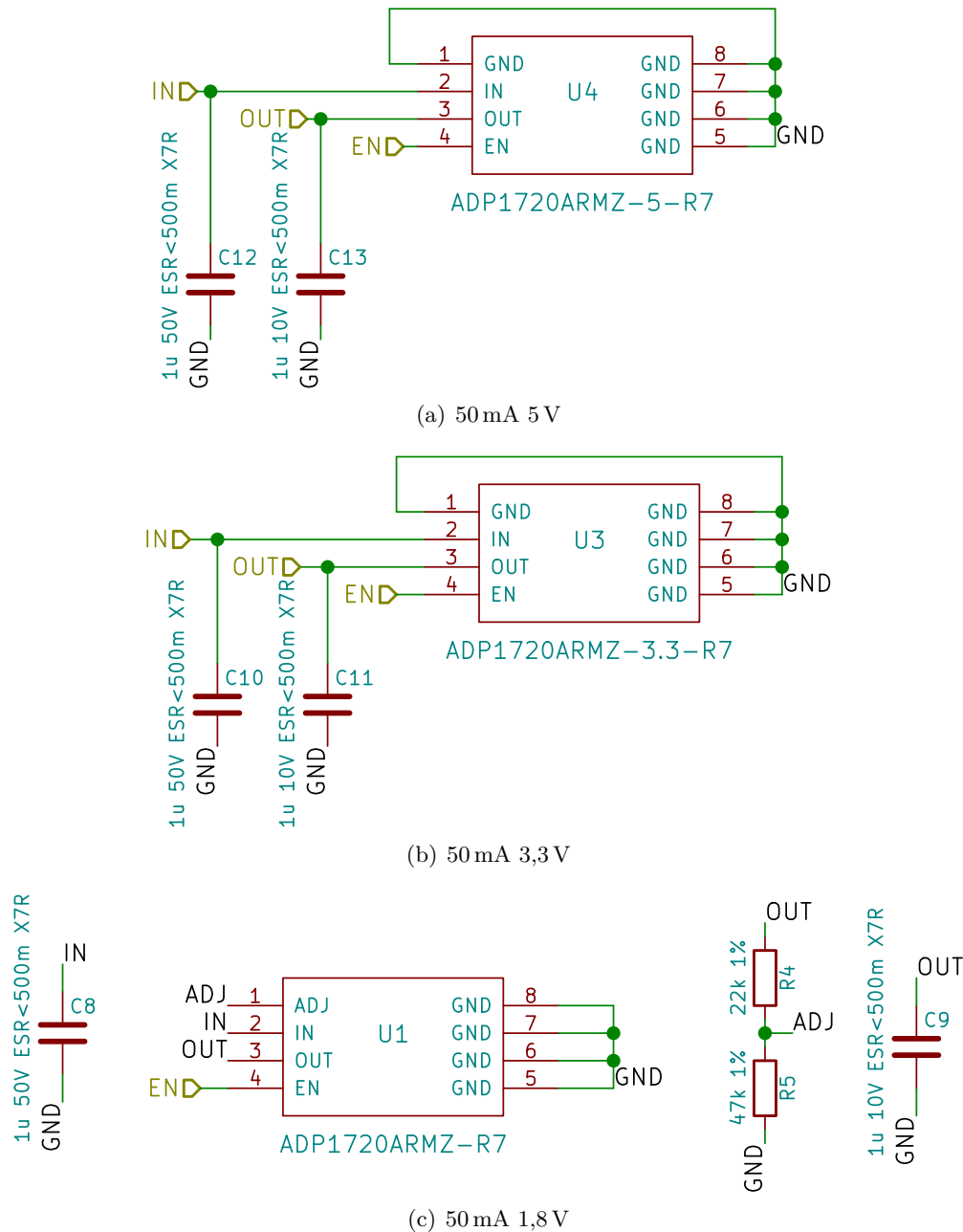
Lineêre spanningsreguleerders word gebruik om lae-ruis toevoerspannings aan die versyferaar, voorversterkers en mikrofone te bied. Lineêre reguleerders is minder effektief as kapperbane, maar omdat die stroomvereistes van die versyferaar, voorversterkers en mikrofone baie laag is, is dit nie 'n probleem nie. Figuur 3.8(a) toon die 5 V-bron, wat gebruik sal word om die analoog-kant



Figuur 3.6: Gedrag van die 3 A 5 V-bron in Figuur 3.5 ten opsigte van die **uitsetstroom**. Die **uitsetspanning** bly binne 0,1 V van die nominale waarde. Beide die effektiwiteit en die ruis neem toe saam met die **uitsetstroom**.



Figuur 3.7: Gedrag van die 3 A 5 V-bron in Figuur 3.5 ten opsigte van die **insetspanning**. 'n Minimum **insetspanning** van ongeveer 7,5 V word verlang.



Figuur 3.8: Lineêre reguleerders om lae-ruis toevoerspannings aan analoogkomponente te verskaf.

van die versyferaar (§5.1), die voorversterkers (§5.2) en die operasionele versterkers (§5.3) aan te dryf. Figuur 3.8(b) toon die 3,3V-bron, wat die digitale kant van die versyferaar (§5.1) aandryf. Die bron in Figuur 3.8(c) gebruik 'n verstelbare weergawe van die ADP1720-reguleerder om 1,8V te skep, wat deur die mikrofone (§4) gebruik sal word.

3.4 Borduitleg en -vervaardiging

Die 5V kapperkragbron, die drie lineêre kragbronne en die *intree*beskerming is saam op een bord geplaas.

Volgens Lengauer (2012) is die outomatiese optimering van stroombaanuitleg 'n moeilike probleem in *niet-deterministiese polinoomtyd* (NP). Dit beteken dat die probleem waarskynlik¹

¹ Solank as wat daar nie 'n bewys gevind word dat $P = NP$ nie.

ten minste so moeilik is soos die berugte probleem van die ‘reisverkoper’ of ‘boodskapper’ wat wil weet wat die optimale roete is waarlangs hy alle stede in die kortste tyd kan besoek (Schrijver, 2005, §7, p.38 ev). Eerder as om die hele oplossingsruimte vir die beste moontlike oplossing te fynkam, word hierdie klas van probleme gewoonlik met heuristiese tegnieke opgelos, wat oplossings bied wat ‘waarskynlik naby’ die optimale oplossing is. Om hierdie rede word die uitleg van stroombaanborde vir projekte soos hierdie gewoonlik met die hand gedoen. Dit is ’n iteratiewe, intuïtiewe en tydrawende proses.

Texas Instruments (2014, §11) maak verskeie aanbevelings vir die uitleg van komponente en spore vir die kapperkragbron:

- Maak spore wat baie stroom dra so wyd as moontlik om weerstand te verminder en hitteuitruiling te verbeter.
- Plaas kapasitors so naby as moontlik aan die TPS563209DDCT-skyfie.
- Plaas vias na ’n grond (GND)-vlak naby die GND-been van elke kapasitor om die weerstand te minimeer en seker te maak dat groot strome nie met ’n ompad sal vloei nie.
- Om die uitstraling van elektromagnetiese ruis te verminder, moet die spoor van die SW-pen kort gehou word. Hierdie spoor moet ook nie onderdeur die TPS563209DDCT-skyfie loop nie.
- Aparte spore moet gebruik word om die uittree aan a) die terugvoerweerstand en b) die las te koppel.
- Om ruis in die terugvoernetwerk (VFB) te verminder, moet die spore van die terugvoernetwerk so kort as moontlik wees en nie naby die spoor van die SW-pen loop nie.

Hierdie aanbevelings is noukeurig gevolg, en die finale uitleg is in Figuur 3.9 te sien, regs van die olifant. Oorbrugging JP3 koppel die EN pen van die TPS563209DDCT-skyfie aan die batteryspanning of aan GND, om die kapperkragbron aan of af te skakel. Oorbrugging JP7 verbind die 5 V-uittree van die kapperkragbron aan die 5 V-pen van die RPI (penne 2 en 4 van ry P1).

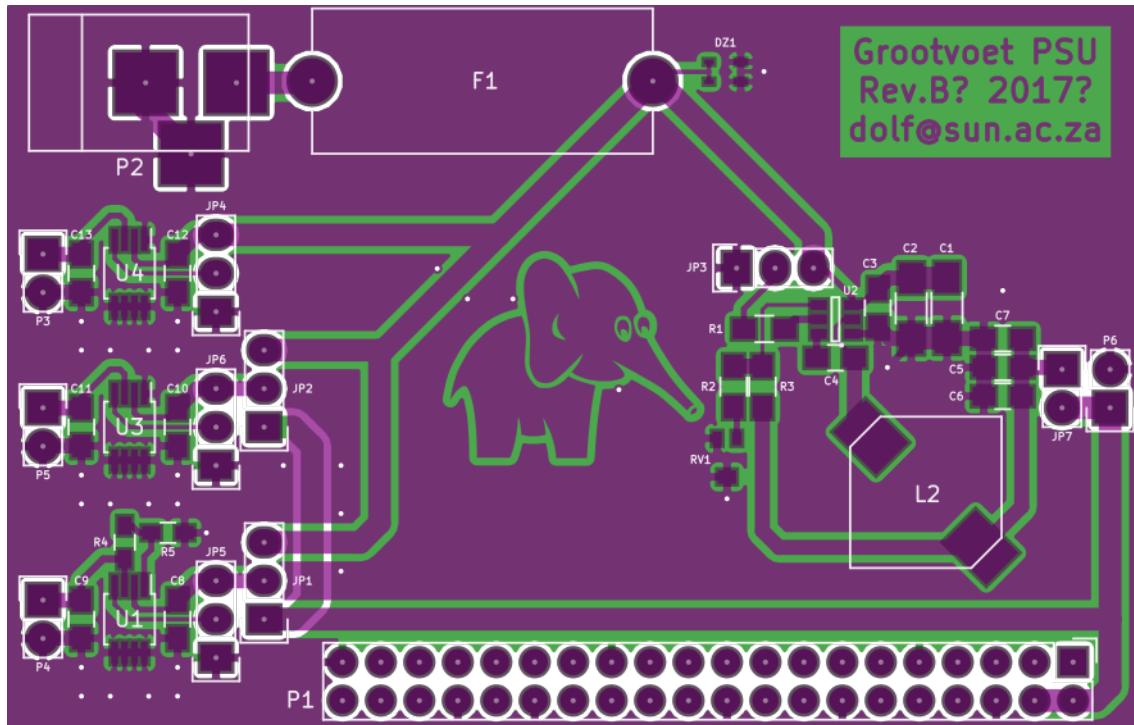
Die lineêre reguleerders se uitleg is eenvoudig. Al die afvlakkapasitors is so na as moontlik aan die relevante ADP1720-skyfie geplaas. Vias na die onderste GND-vlak word naby alle penne wat aan GND verbind is, geplaas. Oorbruggings JP4, JP5, JP6 moet in die boonste posisie geplaas word om die EN-penne van die onderskeie ADP1720-skyfies hoog te trek om hulle aan te skakel. In die onderste posisie koppel dit die EN-penne aan GND, wat die onderskeie lineêre bronne afskakel. JP1 en JP2 word gebruik om te kies of die toevoerspanning van die 3,3 V- en 1,8 V-reguleerders van die uittree van die kapperkragbron (onderste posisie), of van die battery (boonste posisie) geneem word.

Die finale ontwerp is na PCBWay gestuur vir vervaardiging. Figuur 3.10 toon die resultaat. Daarna is alle komponente met die hand gesoldeer.

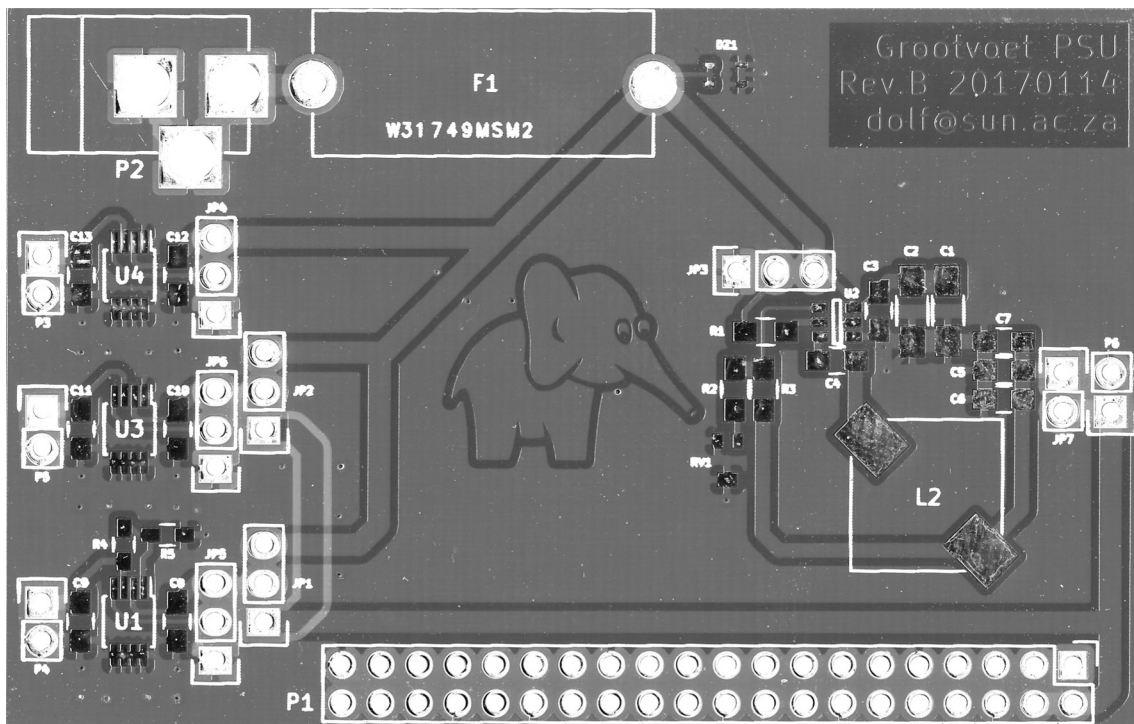
3.5 Kragbronbeheerder

Hierdie projek maak van ’n RPI gebruik. Die RPI het nie ’n slaap-modus nie. Nadat die bedryfstelsel gestop is (met behulp van `sudo poweroff`), gebruik die RPI nog steeds heelwat stroom. Die stroombaan in Figuur 3.11 los hierdie probleem op.

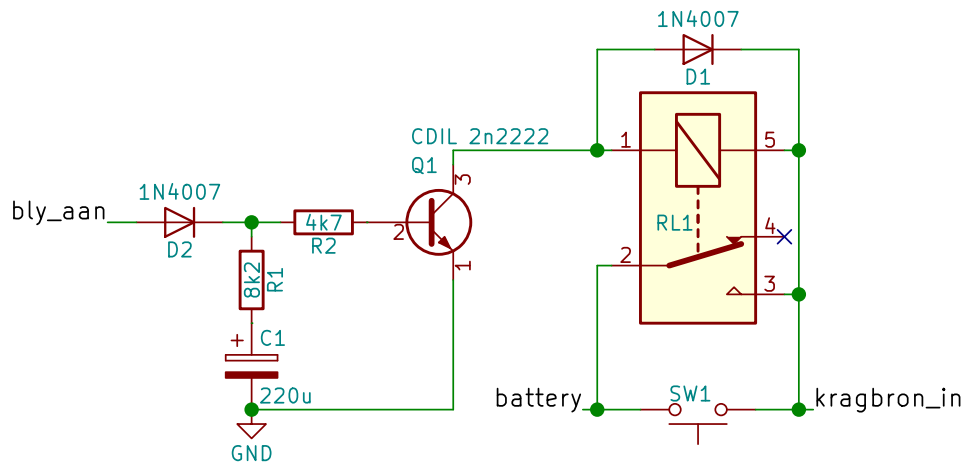
Pen bly_aan word aan ’n GPIO-pen van die RPI gekoppel en moet hoog gehou word solank die kragbron veronderstel is om aangeskakel te bly. Dit word gedoen met behulp van die `gpio -poweroff` ‘Device Tree Overlay’ (DTO). Dit veroorsaak dat die RPI ’n sekere GPIO-pen so



Figuur 3.9: Die borduitleg vir die kragbron is met KiCad ontwerp. Links bo is die batterysok en regs daarvan die sekering en Zener-diode. Die lineêre reguleerders is die drie groepe komponente aan die linkerkant van die bord. Die groep komponente regs van die olifant is die 5 V kapperkragbron. Ry P1 (onder) word gebruik om hierdie bord aan 'n **RPI** te koppel.



Figuur 3.10: Vervaardigde kragbron-stroombaanbordjie. Hierdie figuur is saamgestel deur albei kante van die bord te skandeer voordat enige komponente daarop gesoldeer is en dit met behulp van fotoredigeringsagteware (GIMP) bo-oor mekaar te plaas.



Figuur 3.11: Die kragbronbeheerder. As skakelaar SW1 gedruk word, word die battery aan die *intree* van die kragbron verbind wat die 5 V aan die *RPI* voorsien. Die *Linux*-kern op die *RPI* trek 'n *GPIO*-pen wat aan *bly_aan* gekoppel is, hoog. Dit laat kapasitor C1 laai, en transistor Q1 dryf 'n stroom deur die spoel van relê RL1. Die relê skakel oor en verbind pen 2 aan pen 3, sodat die gebruiker SW1 kan laat los. As die *RPI* afgeskakel word, laat sak die *Linux*-kern die spanning by *bly_aan*. As gevolg van diode D2 ontlaaie die kapasitor stadig deur R1, R2 en die B-E-vlak van die transistor. 'n Paar sekondes later is die kapasitor ontlaaie, en die relê ontkoppel die battery van die res van die stelsel.

gou as moontlik na aanskakeling hoogtrek – en weer laagtrek sodra die bedryfstelsel 'n *halt*-modus betree. Die konsep van 'Device Trees' (DTs) in die *Linux*-kern word breedvoeriger in §7.2 bespreek.

Daar is 'n effense vertraging tussen die *GPIO*-pen wat laaggetrek word en wanneer dit veilig is om die toevoerspanning van die *RPI* te ontkoppel. Kapasitor C1 veroorsaak dat transistor Q1 vir 'n paar sekondes voorgespan bly nadat pen *bly_aan* laag word. Diode D2 voorkom dat stroom uit kapasitor C1 terugvloei na die *RPI*, wat sou veroorsaak dat die battery te gou ontkoppel word. Om die vertraging te verleng, kan C1 en R1 vergroot word.

Diode D1 beskerm die transistor teen hoë spannings wat as gevolg van die induktansie van die spoel opgewek word wanneer die stroom skielik ophou vloei. Om geskik te wees vir hierdie stroombaan, moet die relê 'n spoel hê wat by $12\text{ V} - V_{\text{CE,sat}}$ sal aanskakel, waar $V_{\text{CE,sat}}$ die spanningsval oor die aangeskakelde (versadigde) transistor is.

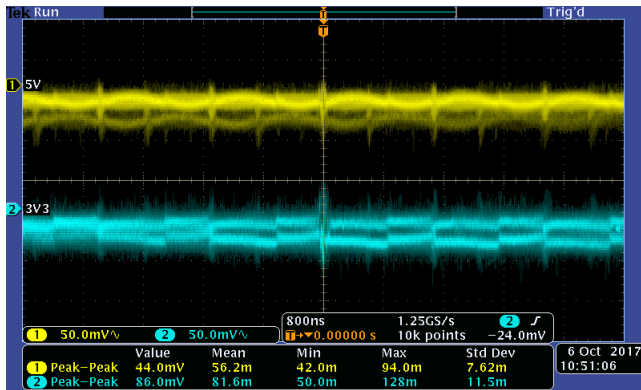
3.6 Ruis met 'n aktiewe las

Kragbronne is nie ideale spanningsbronne nie, maar het eindige *uittree*-impedansies. Dit veroorsaak dat die *uittree*spanning effens daal wanneer stroom gelever word. Hierdie impedansie is ook nie 'n suiwer weerstand nie, maar het 'n induktansie, wat veroorsaak dat die *uittree*spanning styg wanneer die stroomverbruik skielik daal en daal wanneer die stroomverbruik skielik styg. Aktiewe laste bevat transistors wat aan en af skakel en het dus 'n veranderlike stroomverbruik. As gevolg van die induktansie van die kragbronne en die spore tussen die kragbronne en die laste, word ruis op die toevoerspannings geïnduseer.

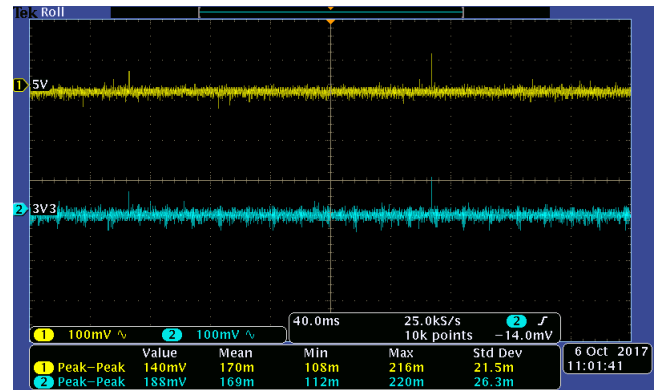
Nadat die projek klaar gebou is, is die effek van die versyferaar in §5, die mikrobeheerder in §6 en die *RPI* in §7 op die kragbronne ondersoek. Figuur 3.12 toon die resultate hiervan. Die mikrobeheerder veroorsaak die meeste ruis, en daardie ruis is op al die toevoerspannings sigbaar – selfs op die battery.

In 'n verdere ondersoek na maniere om hierdie ruis te beperk, is drie verskillende kragbron-konfigurasies tydelik gebou, en die versyferingsruis in elke geval gemeet:

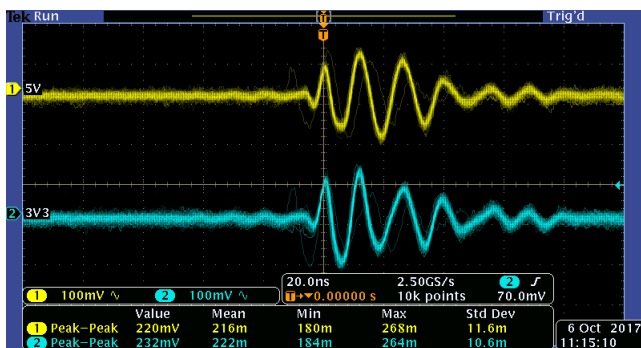
Kapper: Gebruik die 5 V-kapperkragbron wat in §3.2 beskryf is (in plaas van die lineêre 5 V-bron) en die 3,3 V-lyn van die *RPI* (in plaas van die lineêre 3,3 V-bron) om die audio-



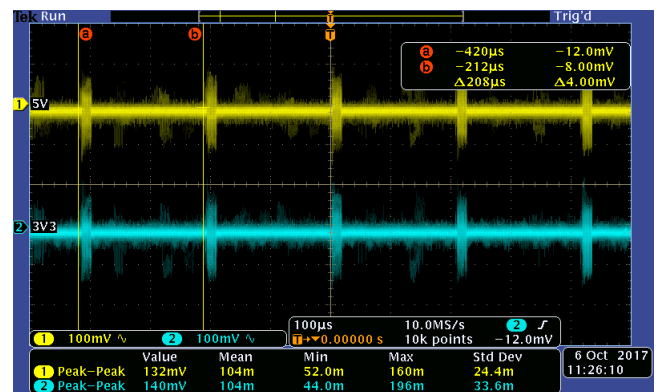
(a) Die ruis op die 5 V-kapperkragbron en die interne 3,3 V-kragbron van die RPI is onderskeidelik 56,2 mV_{pp} en 81,6 mV_{pp} wanneer die RPI die enigste las is. Die mikrobeheerder, versyferaar, GPS en die drie lineêre reguleerders is afgeskakel tydens hierdie toets.



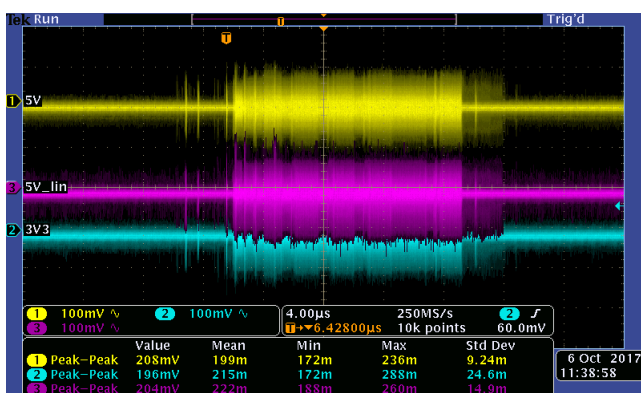
(b) Met die lineêre kragbronne en RPI aangeskakel, maar die res afgeskakel, vermeerder die ruis na ongeveer 170 mV_{pp} op beide lyne.



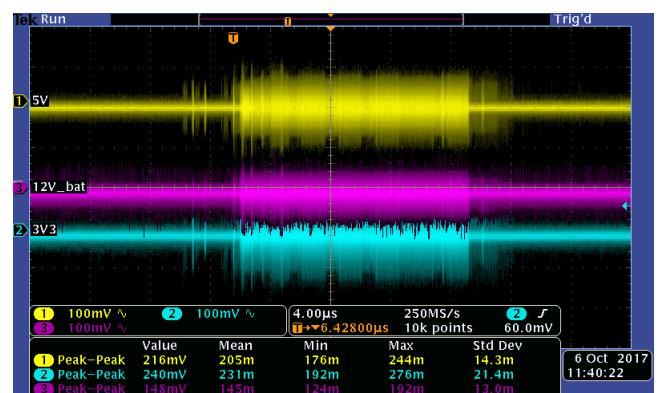
(c) Elke 52,038 μ s vind daar 'n ossilasie met 'n piekspanning van ongeveer 220 mV_{pp} plaas. Hierdie periode stem ooreen met 19,2 kHz, wat die versterk **monstertempo** van die versyferaar is. Dit is dus veilig om aan te neem dat die stroomverbruik van die versyferaar vir hierdie ossilasie op die toevoerspanning verantwoordelik is.



(d) Wanneer die mikrobeheerder aangeskakel word en die program wat met die versyferaar kommunikeer geloop word, vermeerder die ruis drasties. Daar is ook 'n duidelike periode van ongeveer 208 μ s, wat ooreenstem met die 4800 Hz **monstertempo** wat in hierdie projek gebruik word.

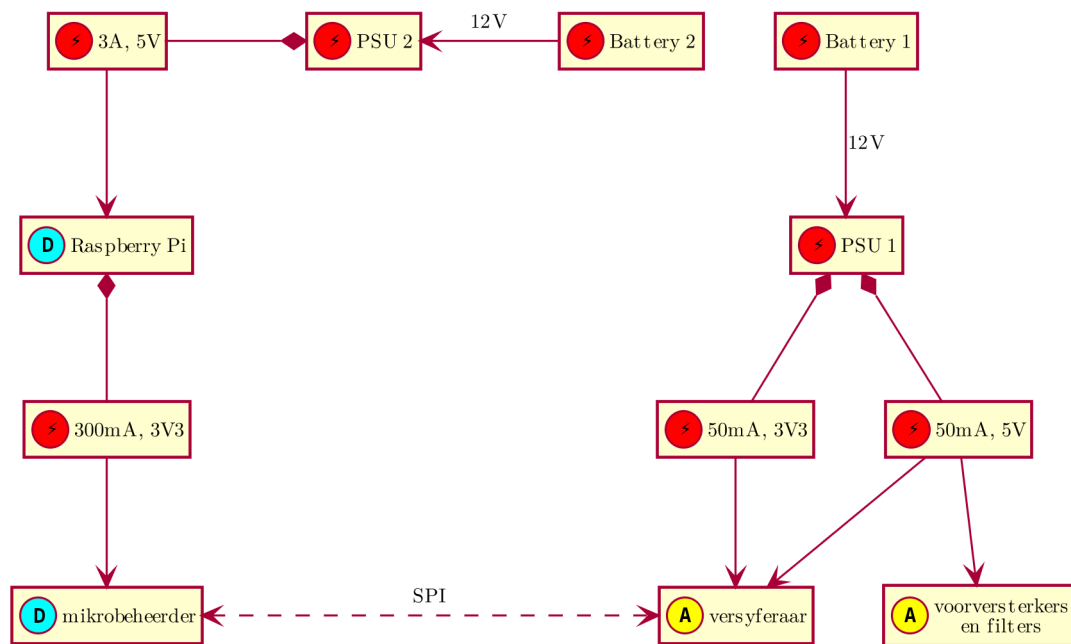


(e) Wanneer die versyferaar en mikrobeheerder loop, koppel die ruis ook met al die lineêre reguleerders. Slegs die 5 V se **uittree** word hier getoon.



(f) Dieselfde ruis is ook op die 12 V toevoerspanning van die loodsuurbattery te sien. Hierdie ruis is dus deur die hele stelsel gekoppel en is 'n aanduiding dat beter ont koppeling met behulp van kapasitors en **ferrietkerne** in die toekoms oorweeg moet word.

Figuur 3.12: Die koppeling van ruis tussen die onderskeie kragbronne is met behulp van 'n Tektronix DPO 3034 ossiloskoop ondersoek. Al die kanale is op **WS**-koppeling gestel, sodat slegs die ruis gemeet word en die **GS**-spanning nie. Die volle bandwydte (300 MHz) van die ossiloskoop is gebruik.



Figuur 3.13: Met hierdie opstelling is probeer om die koppeling van ruis tussen die mikrobeheerder en die oudiokoppelvlak (bestaande uit voorversterkers, filters en 'n versyferaar) te beperk. Die digitale kant van die stroombaan word sover moontlik van die analoogkant geskei deur twee aparte kragbronne en twee aparte batterye te gebruik.

koppelvlak aan te dryf. Dit sou na verwagting die meeste versyferingsruis veroorsaak, maar blyk slegs die versyferingsruis by hoë frekwensies te vermeerder.

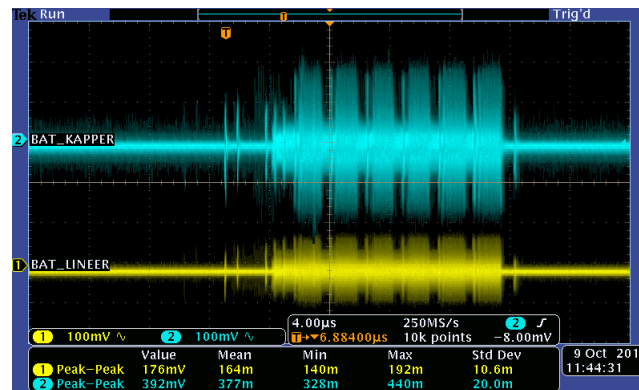
Lineêr: Gebruik die 5 V en 3,3 V lineêre bronne vir die oudiokoppelvlak, terwyl die mikrobeheerder vanaf die RPI se 3,3 V-lyn aangedryf word. Al die kragbronne is dus op dieselfde stroombaan soos in Figuur 3.9. Dit is die opstelling wat in die finale prototipe vir hierdie projek gebruik is. Dit sou na verwagting minder ruiskoppeling veroorsaak as die vorige konfigurasie, maar dit blyk egter slegs die ruis by hoë frekwensies effens te beperk.

Twee batterye: Gebruik twee kragbronne wat aan twee aparte loodsuurbatterye gekoppel is, soos beskryf deur Figuur 3.13. Sodoende is die enigste verbinding tussen die oudiokoppelvlak (§5) en die digitale stroombane (§6 en §7) deur die **serie-koppelvlak vir randapparatuur (SPI, Serial Peripheral Interface)**-bus wat data van die versyferaar na die mikrobeheerder oordra. Figuur 3.14 toon dat die ruis wat deur die mikrobeheerder veroorsaak word in hierdie geval nog steeds aan die tweede battery oorgedra word. Vervolgens bied dit geen verdere verbetering bo die vorige opstelling nie.

Vir elke konfigurasie is die **intreepoorte** van die versyferaar óf na **GND** gekortsluit (0Ω), óf met $200\Omega \pm 1\%$ weerstande aan **GND** gekoppel om die **uittreeweerstand** van 'n die mikrofone in §4 te simuleer. Die resultate word in Figure 3.15(a) en 3.15(b) getoon. Dit blyk dat die gebruik van lineêre bronne en twee aparte batterye nie die ruislvlak verlaag soos verwag nie. Die ruislvlak in die gewenste frekwensie-omvang is in elke geval tussen ongeveer -80 dBFS en -97 dBFS, wat beteken dat slegs 14 van die 24 bisse ruisloos versyfer word.

3.7 Gevolgtrekking

Die ontwerpde kragbron is voldoende vir gebruik in hierdie prototipe, maar laat ruimte vir verbetering ten opsigte van ruis. Dit is onduidelik hoeveel van die versyferingsruis (Figuur 3.15(a)) deur

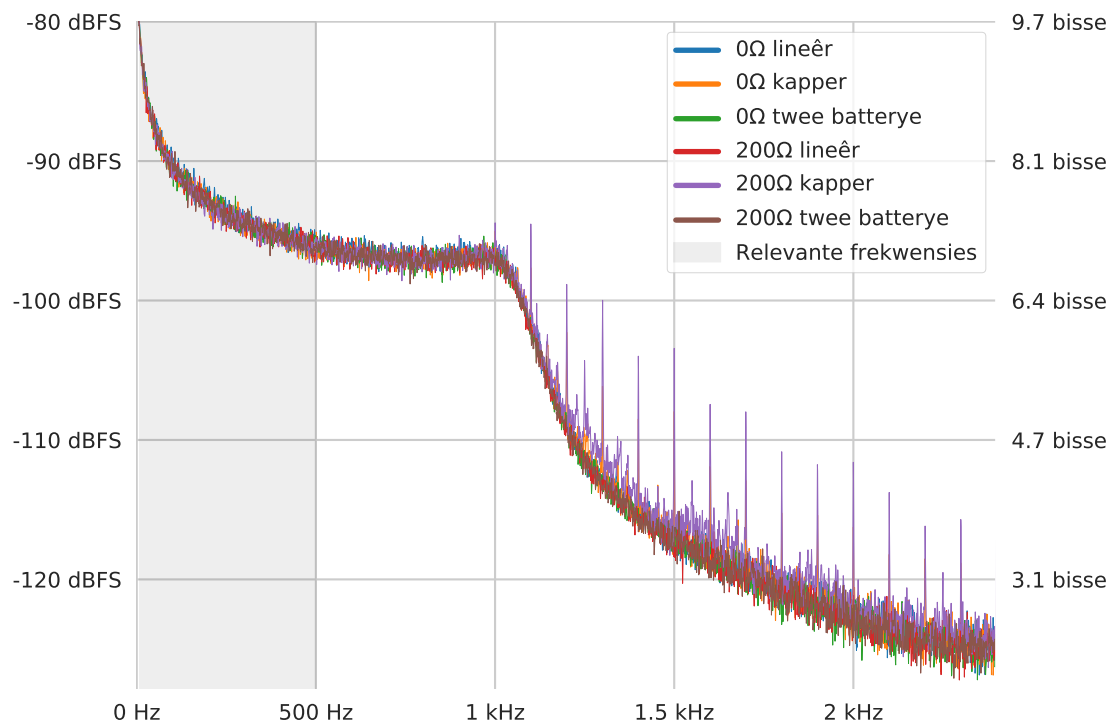


Figuur 3.14: Ondanks die aparte batterye en aparte kragbronne in die opstelling in Figuur 3.13, is die ruis steeds aan albei batterye (en dus ook aan die toevoerspanning van die versyferaar) gekoppel.

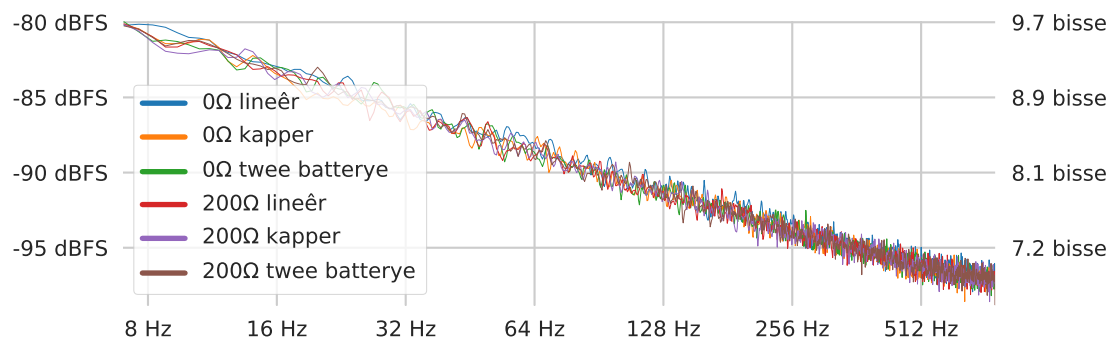
die kapperkragbronne veroorsaak word en hoeveel aan die mikrobeheerder (sien Figuur 3.12(d)) te wyte is. Indien die ruis in die kragbron self nie verminder kan word nie, kan 'n kombinasie van die volgende verbeteringsvoorstelle ondersoek word om die **sein-na-ruis-verhouding (SNR)** van die versyfering te verbeter:

- Isoleer die laste beter deur groter of beter **afvlakkapasitors** by elke kragbron en elke las te plaas en **ferrietkerne** op strategiese plekke te plaas om die ruis tot die digitale gedeelte van die stroombaan in te perk.
- 'n Digitale isoleerder soos die IS07760 van **Texas Instruments (TI)** kan ook oorweeg word om die kommunikasie tussen die versyferaar en mikrobeheerder te isoleer. Dit skep die moontlikheid om twee aparte grondvlakke vir die analoogstroombane en digitale stroombane te gebruik, wat die ruis verder sal beperk.
- Ondersoek kan ook ingestel word na die beskerming van die analoogstroombane (§5) teen ruis wat deur die res van die stroombaan uitgestraal word met behulp van koperplate wat aan **GND** gekoppel is (d.w.s. 'n Faraday-hok).

Die ontwerp van die laste (naamlik die voorversterkers, filters, versyferaar en mikrobeheerder) waarmee die kragbronne getoets is, word in die volgende hoofstukke behandel.



(a) Die versyferingsruis vir die volle frekwensie-omvang van die versyferaar, met 'n **monstertempo** van 4800 Hz. As 'n kapperkragbron gebruik word om die oudiokoppelvlak aan te dryf, is daar meer ruis by frekwensies bo 1000 Hz. Hierdie pieke in die ruispektrum word groter as die **uittree**-impedansie van die mikrofoon in berekening gebring word.



(b) Die versyferingsruis vir die relevante frekwensie-omvang van olifantgeluide, naamlik 8–500 Hz. Die gebruik van lineêre reguleerders of twee aparte batterye maak geen impak op die versyferingsruis in die gewenste omvang nie.

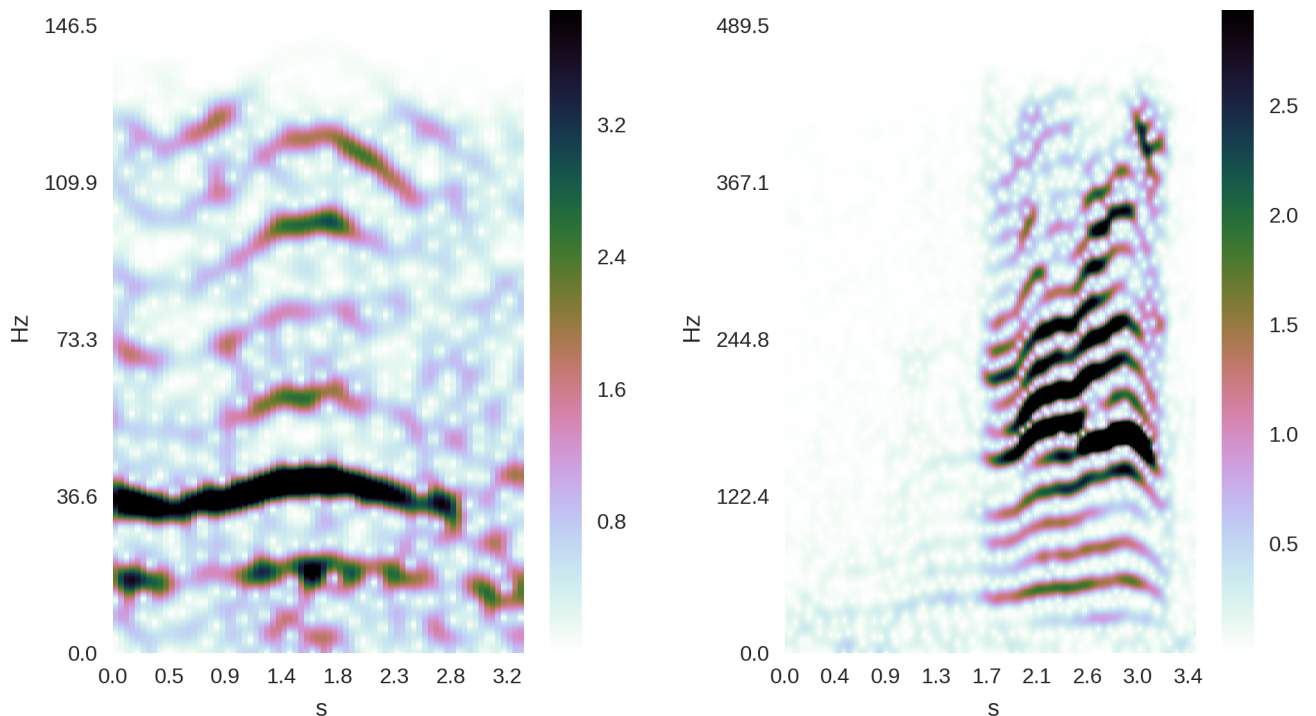
Figuur 3.15: Resultate van die ondersoek na verskeie opstellings om die versyferingsruis te probeer verminder. Die gebruik van lineêre bronne bied slegs 'n verbetering by hoë frekwensies wat buite die frekwensie-omvang van belang is. In beide figure stem 0 dBFS ooreen met die maksimum versyferde waarde. 'n Ruisvlak van -80 dBFS stem dus ooreen met ongeveer 9,7 bisse ruis, wat beteken dat slegs die 14 mees beduidende bisse van die versyfering ruisvry is.

Hoofstuk 4

Mikrofone

Olifantdreungeluide is breë-band klanke, tipies tussen 0,5 s en 5 s lank, met 'n fundamentele frekwensie tussen 15 Hz en 35 Hz (Zeppelzauer *et al.*, 2013). Groot bulle se vokaliserings kan 'n fundamentele frekwensie hê tot so laag as 8 Hz (Zeppelzauer *et al.*, 2014). Olifantkoeie en -kalfies se geluide is effens hoër. Die geluide bevat gewoonlik baie harmonieke. Afhangend van die tipe geluid, kan die harmonieke tot by ongeveer 500 Hz strek. As voorbeeld hiervan, toon Figuur 4.1 spektrogramme van twee olifantdreungeluide uit die datastel van Zeppelzauer *et al.* (2013).

Verskeie tipes vibrasiesensors bestaan, waaronder mikrofone, geofone en versnellingsmeters. Versnellingsmeters is geskik vir die meting van grootskaalse bewegings of vibrasies, terwyl die sensor aan die bewegende voorwerp vasgemaak is en is dus nie geskik vir sagte klanke nie.



(a) 'n Lae dreuning met 'n fundamentele frekwensie van ongeveer 18 Hz en baie lae formante. Hierdie dreuning sal vir die meeste mense skaars hoorbaar wees.

(b) 'n Harmoniese dreuning (1,7–3,4 s) met 'n lae, maar swak, fundamentele frekwensie en sterk formante rondom 150–250 Hz, wat vir 'n mens duidelik hoorbaar sal wees. 'n Sagte, lae dreuning (0–1,7 s) gaan gereeld so 'n dreuning vooraf.

Figuur 4.1: Spektrogramme van twee tipes olifantdreungeluide. Let op die verskillende skale van die frekwensie-asse.

Geofone is soos kontakmikrofone vir die grond en het normaalweg 'n baie lae hoogafsniefrekwensie¹. Mikrofone is geskik om vibrasies in lugdruk te meet en is dus waarskynlik die beste tipe sensor om olifantdreungeluide mee te meet.

4.1 Mikrofoonkeuse

Zeppelzauer *et al.* (2014) noem dat hulle twee mikrofone gebruik het om geluide van mak olifante op te neem:

- 'n AKG C480 B CK 69-ULS, met 'n frekwensieweergawe van $\pm 0,9$ dB oor 8 Hz–20 kHz.
- 'n Alsydige Neumann-mikrofoon wat aangepas is om lae-frekwensie klanke tot by 5 Hz te kan meet.

Die opnemer wat gebruik is, is 'n Sound Devices 722 draagbare opnemer met 'n ingeboude hardeskyf. Teen \$1000 vir 'n enkele AKG-mikrofoon² en \$2699 vir die opnemer³ is dit egter nie 'n bekostigbare opsie vir hierdie projek nie.

Die absolute SPL van tipiese olifantdreungeluide is onbekend en moeilik om te meet. Die gewenste sensitiwiteit van die mikrofone wat gebruik moet word, is dus ook onbekend. Weens 'n gebrek aan tyd is die mikrofone nie op regte olifante getoets nie. Toekomstige werk sluit dus in om te bepaal of die sensitiwiteit geskik is.

Die meeste kommersieel beskikbare mikrofone is vir die 20 Hz–20 kHz-omvang bedoel en het dus 'n 3 dB-afsniefrekwensie tussen 20 Hz en 100 Hz. Baie vervaardigers vermeld nie 'n gedetailleerde frekwensieweergawe van die mikrofoon in die datablad nie. Indien die frekwensieweergawe wel gegee word, word die weergawe onder 20 Hz amper altyd verswyg. Die metode waarmee dit gemeet is, word ook selde genoem. Figuur 4.2(a) toon 'n voorbeeld hiervan: Die frekwensieweergawe van die Behringer ECM8000 kondensator-mikrofoon word nie laer as 60 Hz gespesifiseer nie. E-poskorrespondensie met 'n verteenwoordiger van MUSIC Group Research UK Limited het daarop gedui dat die vervaardiger self nie inligting het oor die gedrag van hul eie mikrofone by daardie frekwensies nie. Die Earthworks Audio M50 en Audio Technica AT8010 is van die min mikrofone waarvoor die frekwensiegedrag wel vermeld word (Figure 4.2(b) en 4.2(c)).

Nog 'n uitdaging met die keuse van mikrofone is die spesifikasie van ruis in die datablad. Mikrofoon-ruis word gewoonlik gegee as 'n enkele getal met 'n eenheid dBA en nie as 'n ruispektrum nie. Die 'A' in die eenheid staan vir 'A-weighting'. Dit is 'n kurwe wat meer gewig aan sekere frekwensies gee as aan ander, sodat slegs die ruis wat vir mense hoorbaar is, in ag geneem word (Georg Neumann GmbH, 2015). Die ruis by lae frekwensies, wat juis belangrik is vir hierdie projek, word dus nie in ag geneem nie.

Mikrofone wat infrasoniese klank kan opneem is nie algemeen beskikbaar nie en is gewoonlik duur, omdat dit as spesialistoerusting beskou word. Tabel 4.1 toon 'n lys van mikrofone wat na bewering sensitief is vir lae frekwensies. Ter vergelyking het die populêre Shure SM57 mikrofoon⁴ 'n gespesifiseerde omvang van 40 Hz–15 kHz.

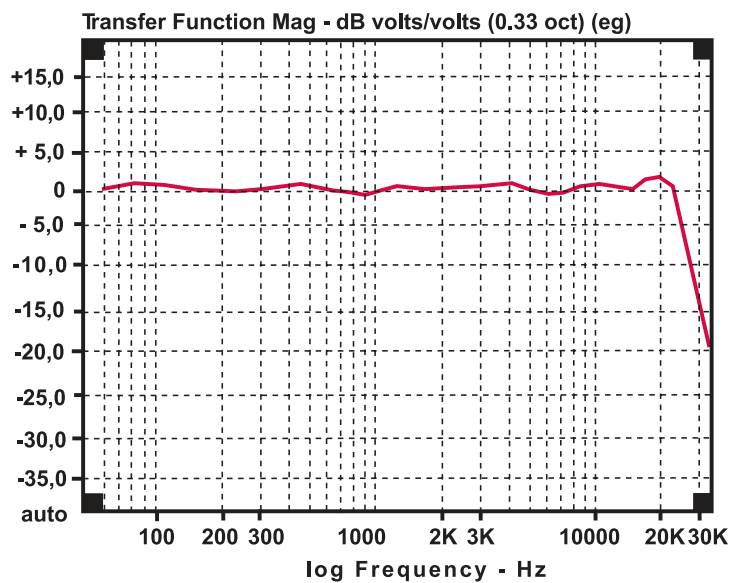
ICS-40300 is 'n goedkoop MEMS-mikrofoon met 'n lae-frekwensie afsnypunt by ongeveer 6 Hz volgens die datablad. Geen ander MEMS-mikrofone is gevind wat so 'n lae afsniefrekwensie het nie. In teenstelling met die meeste ander vervaardigers, verskaf Invensense nie net die verwagte frekwensiegedrag van die mikrofone nie (Figuur 4.3(a)), maar ook die volledige ruisdigtheidspektrum (Figuur 4.3(b)). Soos verwag, is die ruis heelwat meer by lae frekwensies as wat die geweegde

¹ 'n Voorbeeld van 'n geofoon is by <https://www.sparkfun.com/products/11744> beskikbaar. Dit het 'n frekwensie-omvang van 10–240 Hz.

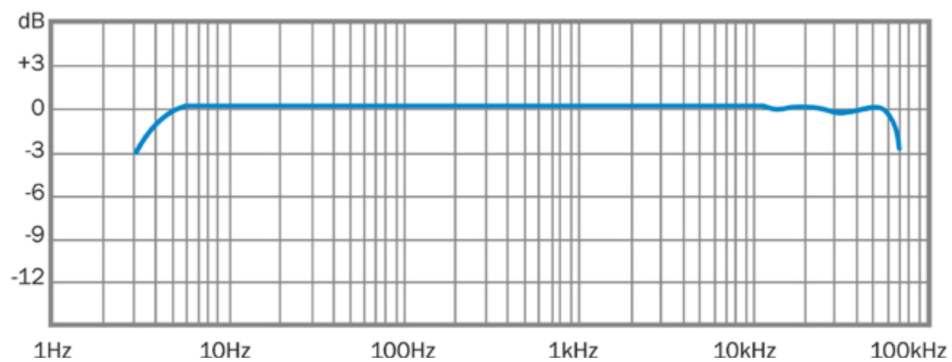
² By <https://www.sweetwater.com/store/detail/C480BCK61> beskikbaar.

³ Hierdie spesifieke opnemer blyk nie meer beskikbaar te wees nie. Sien <https://www.sweetwater.com/store/detail/702T> vir 'n soortgelyke eenheid.

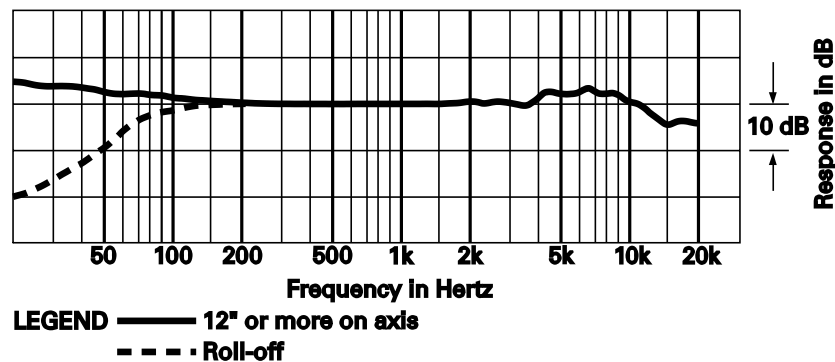
⁴ Beskikbaar teen R1849 by https://www.audiomart.co.za/product_info.php/shure-sm57-dynamic-instrument-microphone-p-5115.



(a) Gespesifiseerde frekwensiegedrag van die Behringer ECM8000 (Figuur uit Behringer (1999) geneem). Geen inligting is beskikbaar omtrent die gedrag van die mikrofoon vir frekwensies laer as 20 Hz nie.



(b) Gespesifiseerde frekwensiegedrag van die Earthworks Audio M50 (Die figuur is uit Earthworks (datum onbekend) geneem). Die frekwensiegedrag is gespesifiseer tot by die -3 dB hoog- en laagafsnypunte. Die laagafsnypfrekwensie is 3 Hz.

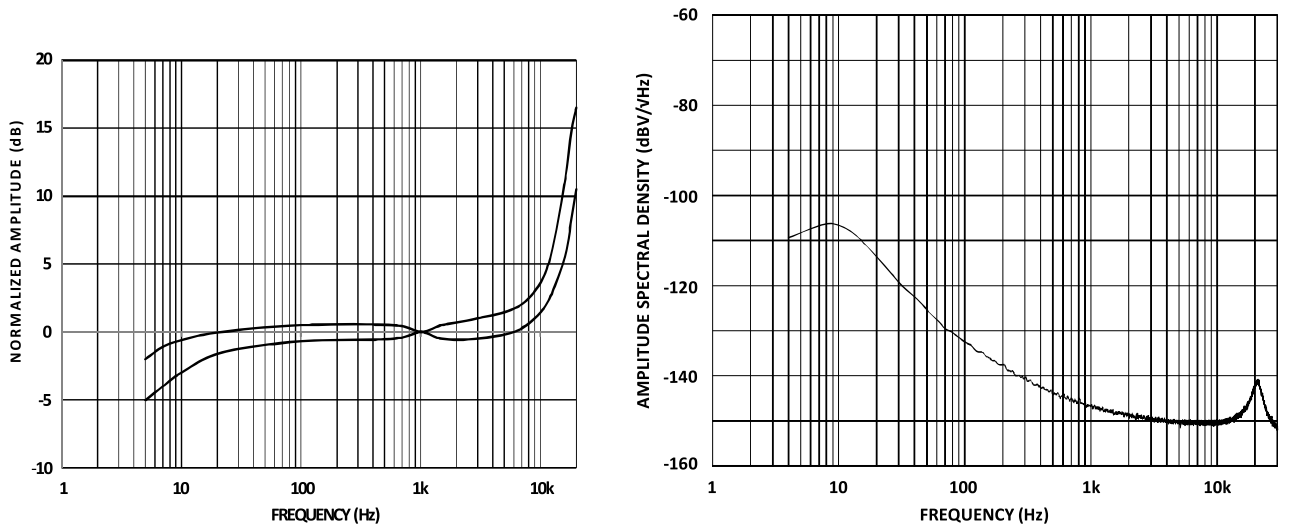


(c) Gespesifiseerde frekwensiegedrag van die Audio Technica AT8010 (Figuur uit Audio Technica (2016) geneem). Hierdie mikrofoon het 'n skakelaar waarmee die hoogdeurlaatfilter afgeskakel kan word. Die frekwensiegedrag word nie gestip vir frekwensies laer as 20 Hz nie.

Figuur 4.2: Frekwensiegedrag van verskeie mikrofone, uit die onderskeie datablaaie oorgeneem.

Vervaardiger	Model	Frekwensiedrag	Beskikbaar by	Prys
Earthworks Audio	M50	3 Hz–50 kHz, +1 dB, –3 dB Sien Figuur 4.2(b).	sweetwater.com/store/detail/M50	R21 k
Brüel & Kjær	mikrofoon 4190-L-001 voorversterker 2669-L	3,15 Hz–20 kHz, ±2 dB	ri@reins.co.za (op aanvraag)	R70 k
BSWA	mikrofoon MK101 voorversterker MV201	3 Hz–16 kHz	meas@spectran.co.za (op aanvraag)	R34 k
Audio Technica	AT8010	Verstelbaar–20 kHz Sien Figuur 4.2(c).	apexpro.co.za/wp-content/uploads/2017/07/Audio-Technica-Pricelist.pdf digikey.com/short/q72n73	R5300
Invensense	ICS-40300	6 Hz–20 kHz Sien Figuur 4.3		R58

Tabel 4.1: Frekwensiedrag en pryse van lae-frekwensie mikrofone. Alle pryse is na ZAR omgeskakel met 'n wisselkoers van 14 ZAR/USD, met **BTW** ingesluit en afgerond. Die **Invensense** mikrofoon is ordegroottes goedkoper as die kapasitormikrofone.



(a) Die frekwensiegedrag is betreklik plat vir die gewenste 8–500 Hz omvang. Die hoë sensitiviteit by hoë frekwensies sal uitgefilter word. Die twee lyne toon die maksimum en minimum groottegewawe – die waarskynlikheidsverdeling tussenin word nie gegee nie.

(b) Soos met die meeste oudiotoerusting neem die ruisdigheidsspektrum toe by laer frekwensies.

Figuur 4.3: Frekwensiegedrag en ruisdigheidsspektrum van die Invensense ICS-40300. Beide figure is uit die mikrofoon se datablad geneem (Invensense Inc., 2015).

dBa-syfer wat gewoonlik gegee word, laat blyk. Wanneer die twee belangrikste vereistes, naamlik koste en frekwensieweergawe, in ag geneem word, blyk dit tog die enigste mikrofoon te wees wat tegelyk bekostigbaar is en 'n voldoende lae afsnyfrekwensie het.

4.2 Hoogdeurlaatfilter

Die datablad beveel aan dat die mikrofoon deur 'n eerste-orde hoogdeurlaatfilter aan 'n voorversterker gekoppel word, sodat die GS-komponent verwyder word. Figuur 4.4 toon die stroombaandiagram van die mikrofoon met die filter. Die afsnyfrekwensie moet heelwat laer as 8 Hz wees, sodat die fundamentele frekwensies van die olifantgeluide nie verswak word nie.

Om 'n geskikte hoogdeurlaatfilter te ontwerp, is kennis van die algoritmes wat gebruik gaan word om die klankbronrigtings te bereken, noodsaaklik: Die kruiskorrelasie tussen die gemonsterde seine van elke mikrofoon word gebruik om die verskil in aankomstyd van 'n sekere sein by twee verskillende mikrofone (die tydvertraging) te bereken. Die posisie van die piek van die kruiskorrelasie stem met hierdie tydvertraging ooreen. Onder die aanname dat die afstand na die klankbron groot is in vergelyking met die spasie tussen twee mikrofone, kan die rigting van die klankbron met behulp van Vergelyking 4.1 bereken word.

$$\alpha = \arcsin \left(\frac{\tau v}{d} \right)$$

waar α = rigting van klankbron,

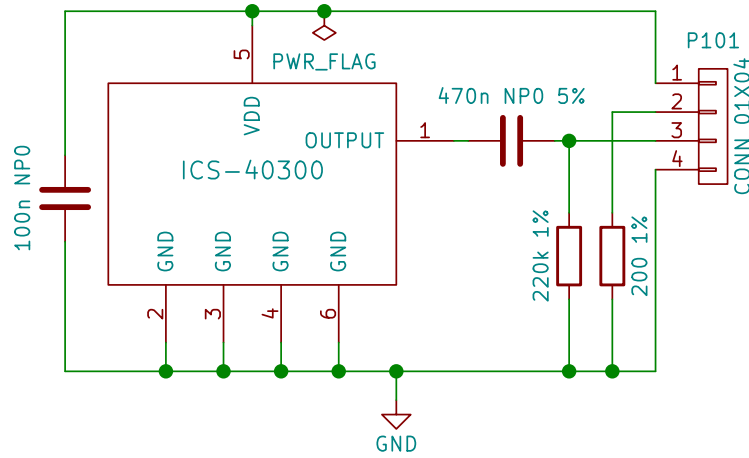
τ = tydvertraging [s],

v = spoed van klank = $343 \text{ m} \cdot \text{s}^{-1}$,

en d = afstand tussen mikrofone [m].

(4.1)

Met 'n **monstertempo** van 4800 Hz is die resolusie van die diskrete kruiskorrelasie gelyk aan die monsterperiode, wat ongeveer $208,3 \mu\text{s}$ is. As die mikrofone byvoorbeeld 20 cm uitmekaar is



Figuur 4.4: Stroombaandiagram van die ICS-40300 met 'n hoogdeurlaatfilter. Vir die 470 nF filterkapasitor word 'n KEMET C2220C474J5GACTU (50 V) gebruik. Vir die 100 nF afvlakkapasitor word 'n TDK FG16C0G2A104JRT06 (100 V) gebruik. Die spanningshantering van die kapasitors word doelbewus hoër gekies as wat nodig is, omdat die kapasitansie van keramiekkapasitors afneem wanneer 'n konstante GS-spanning oor die kapasitor geplaas word. Hierdie verlaging is minder by kapasitors met 'n hoër nominale spanningshantering. Albei kapasitors is van die NP0-tipe, omdat dit beter langtermyn-stabiliteit bied.

en die spoed van klank is $343 \text{ m} \cdot \text{s}^{-1}$, is die resolusie van α hoogstens⁵ $20,9^\circ$. Die kruiskorrelasie moet dus geïnterpoleer word om 'n meer akkurate beraming van τ te kry. Die algoritmes daarvoor word in §10.1 noukeuriger behandel.

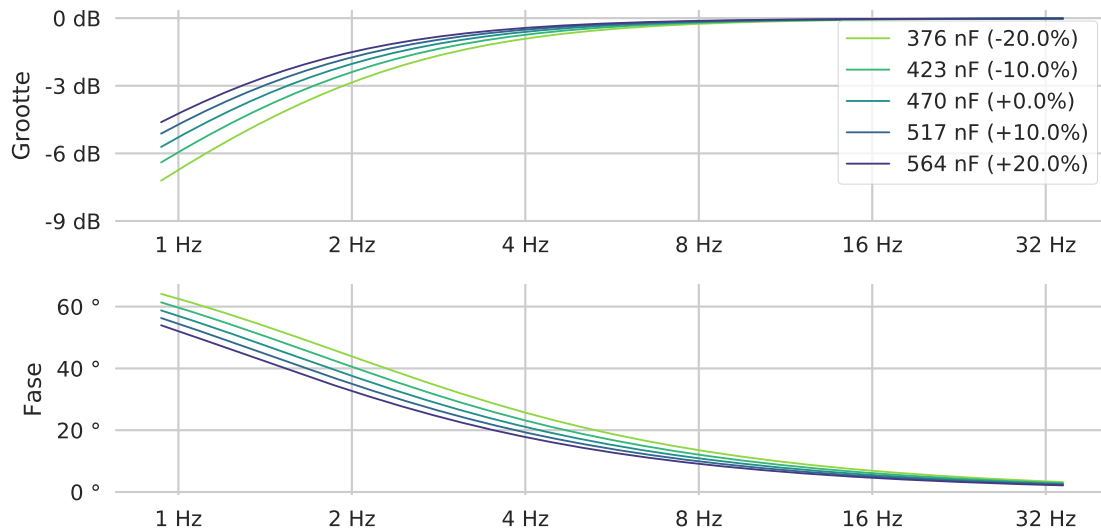
Mikrofone, net soos ander komponente, het 'n sekere toleransie: twee mikrofone uit dieselfde fabriek is nooit identies nie (byvoorbeeld soos aangedui in Figuur 4.3(a)). Hierdie verskille word vererger deur die nie-idealiteit van die hoogdeurlaatfilter waaraan die mikrofoon gekoppel word.

Die filterkomponent met die grootste toleransie is die kapasitor C . Figuur 4.5 toon die effek van die kapasitansie van C op die oordragfunksie van die hoogdeurlaatfilter. Die grootste beskikbare keramiekkapasitor by DigiKey met 'n toleransie van 5%, is 470 nF (KEMET C2220C474J5GACTU). Laer toleransies is nie oor die algemeen beskikbaar nie. Die res van die filter is rondom hierdie kapasitorwaarde ontwerp. Weerstande met 1% toleransie word gebruik, sodat die effek van daardie toleransies weglaatbaar klein is teenoor dié van die kapasitor.

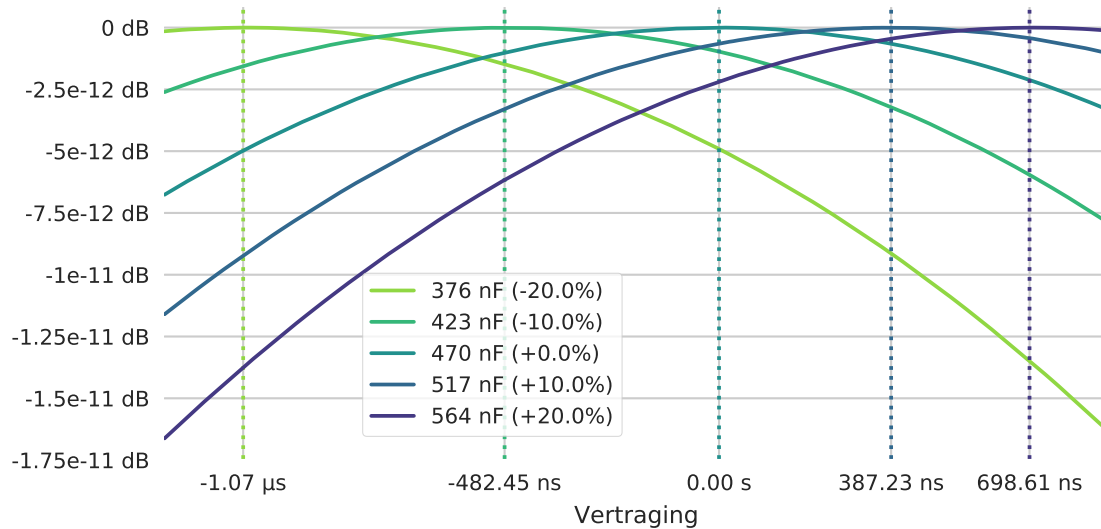
Simulasies is uitgevoer om die impak van die verskillende waardes van C op die kruiskorrelasie te bepaal:

1. $s_{\text{bron}} \leftarrow$ wek ruissein op.
2. $H \triangleq$ ideale hoogdeurlaatfilter (met $C = 470 \text{ nF}$).
3. $s_1 \leftarrow H(s)$.
4. $\mathbf{c} \leftarrow$ verskillende kapasitorwaardes bo en onder die nominale 470 nF.
5. Vir elke c in \mathbf{c} :
 - 5.1. $\tilde{H} \triangleq$ nie-ideale hoogdeurlaatfilter met $C = c$ (soos in Figuur 4.5).
 - 5.2. $s_2 \leftarrow \tilde{H}(s)$
 - 5.3. Bereken kruiskorrelasie tussen s_2 en s_1 .
 - 5.4. Interpoleer kruiskorrelasie en vind die posisie van die piek.

⁵ As gevolg van die nie-lineariteit van arcsin neem die resolusie af soos wat $\left| \frac{\tau v}{d} \right|$ groter word. Sien §10.1 vir meer besonderhede.



Figuur 4.5: Die hoogdeurlaatfilter in Figuur 4.4 word sterk deur die waarde van kapasitor C beïnvloed. Hoe laer die afsnyfrekwensie en hoe kleiner die toleransie van die kapasitor, hoe minder is die fasevervorming van die sein in die gewenste deurlaatband (bó 8 Hz). Hierdie figuur toon die afwyking in frekwensiegedrag vir 'n kapasitor met 20 % toleransie.

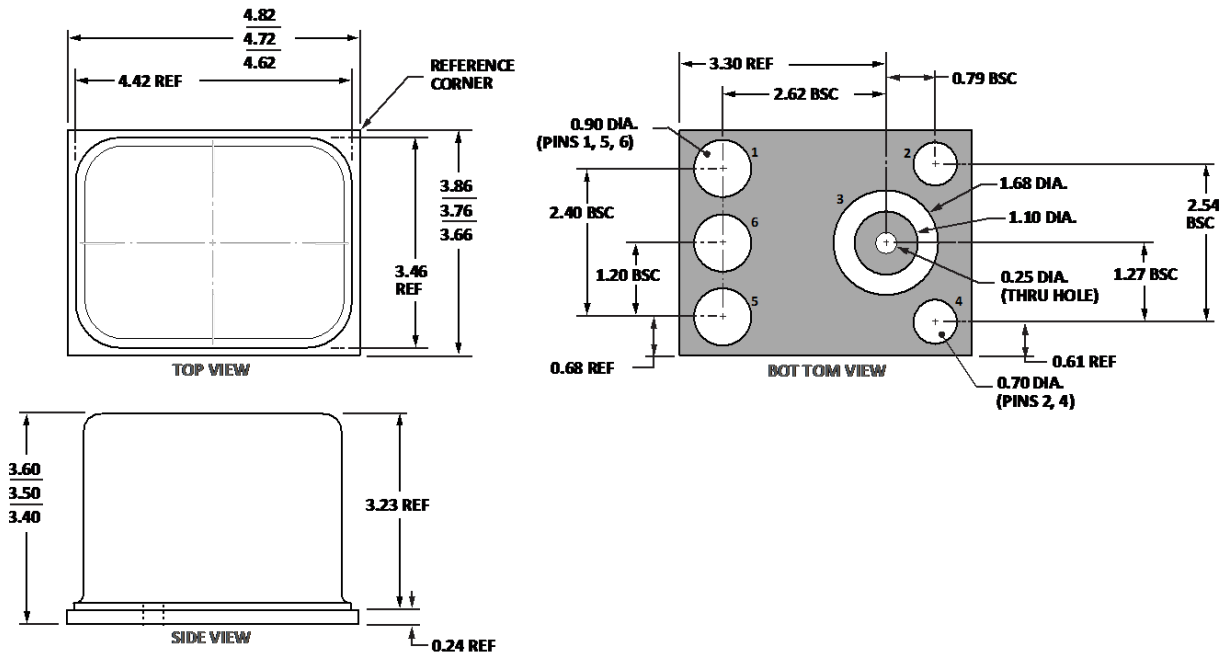


Figuur 4.6: Die pieke van die kruiskorrelasies verskuif effens wanneer die fase van een van die seine vervorm word. Dit veroorsaak 'n hoekfout in die klankbronrigtingberaming.

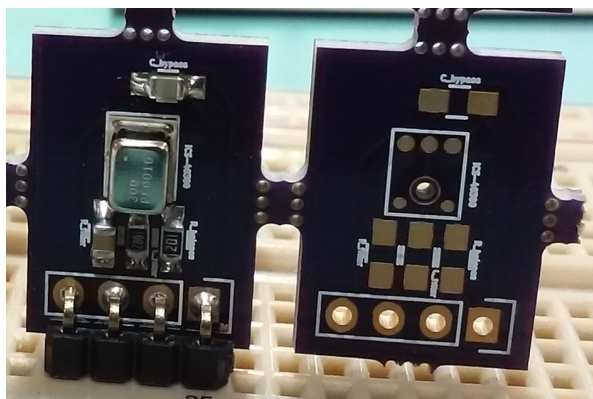
Figuur 4.6 toon die geïnterpoleerde kruiskorrelasies. Die posisies van die pieke word op die absis aangedui. Alhoewel dit lyk asof die verandering in posisie van die kruiskorrelasie se piek baie klein is, is dit nie 'n weglaatbare verandering nie. 'n Afwyking van $1 \mu\text{s}$ in τ stem byvoorbeeld ooreen met 'n fout van $0,0983^\circ$ in α as die mikrofone 20 cm uitmekaar is. Dit sou 'n posisiefout van 1,71 m veroorsaak vir klankbronne wat 1 km ver is.

Die finale⁶ hoogdeurlaatfilter word in Figuur 4.4 aangedui en het 'n afsnyfrekwensie van ongeveer 1,54 Hz.

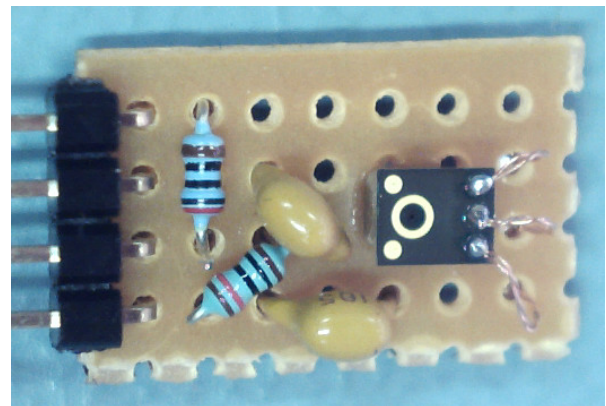
⁶ As gevolg van die nie-linéêre aard van die ontwerpproses het 'n vorige iterasie van die filterontwerp tot die einde vergete gebly. Die fisiese prototipe wat gebruik is vir die toetse in §10 gebruik hoogdeurlaatfilters met afsnyfrekwensies by 16 Hz en kapasitors met 20 % toleransie. Dit is 'n eenvoudige, maar tydrovende taak om die mikrofoonbordjies in Figuur 4.8(b) met die nuwe komponente soos aangedui in Figuur 4.4, te herbou.



Figuur 4.7: Fisiese afmetings (in mm) van die ICS-40300-mikrofoon. Figuur is uit die mikrofoon se datablad geneem (Invensense Inc., 2015).



(a) Professioneel vervaardigde stroombaanbordjies, met (links) en sonder (regs) komponente. Hierdie oplossing het nie gewerk nie. Dit is moontlik dat die soldeerproses die mikrofone beskadig het.



(b) 'n Alternatiewe oplossing wat wel gewerk het. Die mikrofoon is onderstebo aan die bord vasgegom, en dun koperdraadjies is onder 'n vergrootglas met die hand aan drie van die mikrofoon se bene vasgesoldeer.

Figuur 4.8: Stroombaanbordjies vir die ICS-40300-mikrofoon met hoogdeurlaatfilter.

4.3 Stroombaanborde

Die ICS-40300-mikrofoon is baie klein en moeilik om te soldeer. Figuur 4.7 toon die fisiese afmetings van hierdie mikrofoon.

Stroombaanbordjies vir die diagram in Figuur 4.4 is ontwerp en vervaardig. Figuur 4.8(a) toon die vervaardigde stroombaanbordjie. Die soldeerwerk was egter keer op keer onsuksesvol. 'n Alternatiewe oplossing is bedink: Die mikrofone is onderstebo met epoksiegom op 'n stukkie Veroboard vasgeplak. Met spesiale voorsig om elektrostatische ontlading te voorkom, is dun draadjies met die hand aan die onderkant van die mikrofone vasgesoldeer. Op hierdie manier is al die mikrofone suksesvol aan die voorversterkers gekoppel. Figuur 4.8(b) toon een van die suksesvolle eindprodukte.

Die vervaardiger van hierdie mikrofone het verskeie toepassingsnotas gepubliseer wat noukeurig

verduidelik hoe hierdie **MEMS**-mikrofone gemonteer moet word om dit teen stof en vog te beskerm, sonder om die frekwensiegedrag negatief te beïnvloed (Invensense Inc., 2013, 2014). Vir hierdie prototipe word daar nie daaraan aandag gegee nie.

4.4 Windbeskerming

Tydens die eerste buitelugtoetse het dit gou geblyk dat windbeskerming om die mikrofone noodsaaklik is. Sonder windbeskerming veroorsaak 'n ligte briesie reeds genoeg ruis om die versyferaar te laat versadig, sodat die sein afgesny word. Die oplossing is om 'n raampie vir die mikrofoon te maak met 'n 3D-drukker en dit in namaakpels⁷ toe te draai. Figuur 4.9 toon 'n gelyktydige opname deur twee ICS-40300-mikrofone – een met windbeskerming en een daarsonder. Die mikrofone waarmee die opnames gemaak is, word daarlangs getoon.

Al die mikrofone wat vir hierdie projek gebruik word, word op hierdie manier teen windgeraas beskerm.

4.5 Mikrofoonskikking

Om klankbronopsoring te doen, moet die mikrofone in 'n vasgestelde manier gerangskik word. Die relatiewe posisies van die mikrofone moet bekend en akkuraat wees.

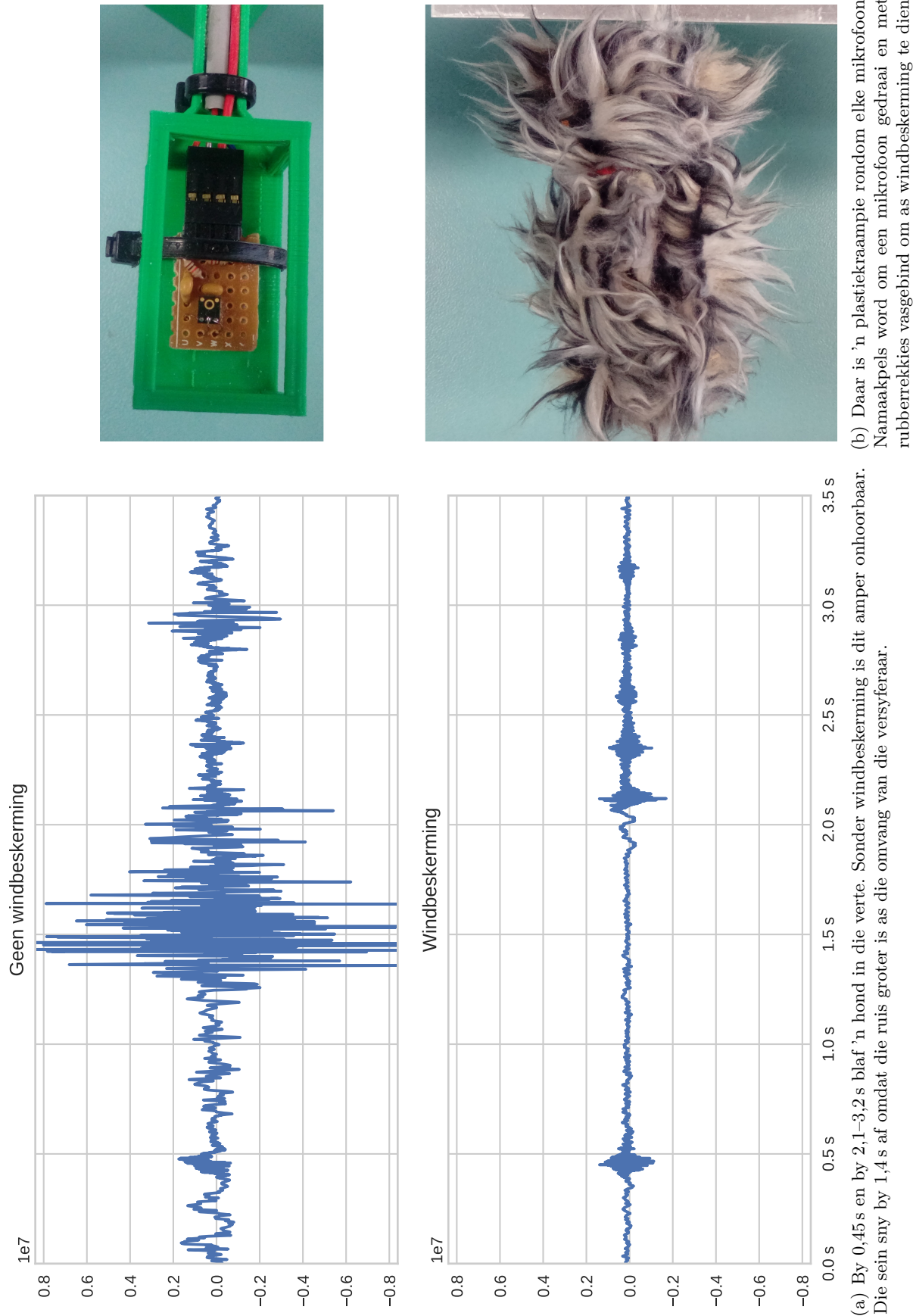
Hoe meer mikrofone per opnemer, hoe meer data is daar om te analiseer en hoe beter is die kanse vir akkurate klankbronopsoring. Siende dat die ICS-40300-mikrofone betreklik goedkoop is, is die enigste beperking op die aantal mikrofone die aantal kanale van die versyferaar wat gebruik word. Die versyferaar wat vir hierdie projek gekies word (sien §5.1) het 6 kanale.

Verskeie skikking-topologieë bestaan, waarvan lineêre skikkings in die literatuur die meeste aandag blyk te geniet. Lineêre skikkings bied geslote-vorm oplossings vir **straalvorming** in die **verrevelde** en is dus geskik vir klankbronopsoringsalgoritmes wat van **straalskandering** gebruik maak. Sulke skikkings is egter nie **isotropies** nie. Vir hierdie projek is dit wenslik dat die opspoorapparaat dieselfde in alle rigtings sal werk, omdat olifante hul ewe waarskynlik aan enige kant van die apparaat kan bevind.

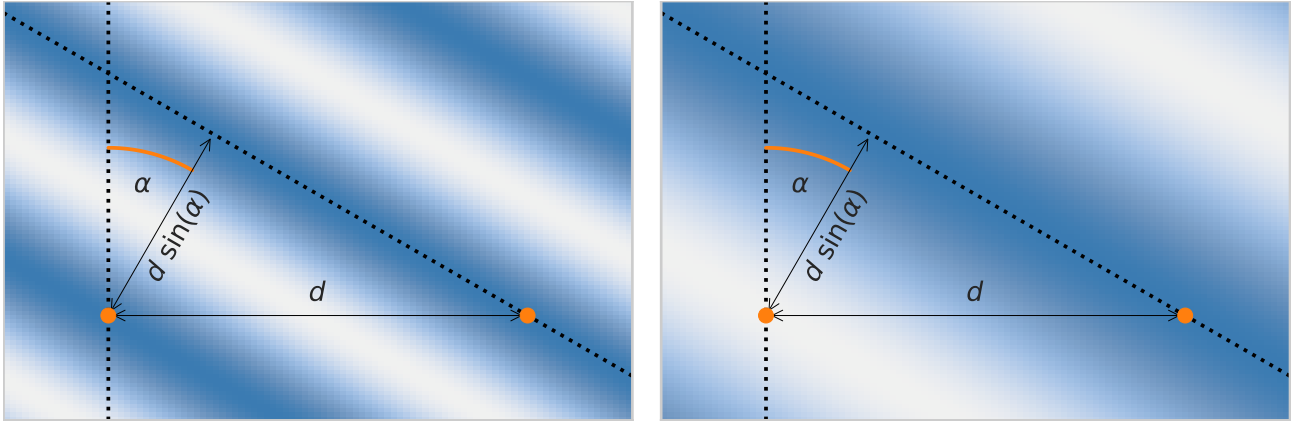
Sirkelvormige skikkings toon wel dieselfde gedrag in alle rigtings, as gevolg van rotasie-simmetrie. As die 6 mikrofone in 'n sirkel gerangskik word, het die skikking 'n sesde-orde rotasiesimmetrie. 'n Nie-lineêre skikking beperk wel die aantal beskikbare algoritmes vir klankopsoring. Vir die doeleindes van hierdie projek is dit toelaatbaar: In §10 sal daar op twee algoritmes gelet word wat die mikrofone in pare beskou en vir elke paar 'n kruiskorrelasie bereken. Sulke algoritmes is toepaslik vir enige topologie. §10.1 beskryf in detail hoe die kruiskorrelasie gebruik kan word om die rigting van die klankbron te bepaal.

Die kruiskorrelasie wat op 'n mikrofoonpaar uitgevoer word, word sterk beïnvloed deur die afstand tussen die twee mikrofone. Figuur 4.10 wys hoe 'n ruimtelike Nyquist-frekwensie $\frac{v}{2d}$ met 'n eenheid van m ontstaan, waar v die spoed van klank is. Die afstand tussen die mikrofone d se komponent in die rigting van die aankomende klankgolffront is $d \sin(\alpha)$ waar α die aankomshoek van die golffront is. Figuur 4.10 toon gevalle waar die frekwensie van die aankomende klankgolf $f = \frac{v}{d \sin(\alpha)}$ (links) en $f = \frac{v}{2d \sin(\alpha)}$ (regs) is. In beide gevalle vind oorvleueling van α plaas. Dit wil sê: Die seine wat deur elke mikrofoon gemeet word vir 'n sekere waarde van α is identies aan die waardes wat gemeet sou word as α 'n ander waarde gehad het. Dit is dus onmoontlik om die rigting van die klankbron te onderskei as so 'n oorvleueling plaasgevind het.

⁷ Kommersiële windbeskermers mag dalk nog beter werk, maar is betreklik duur. Die Røde Deadcat is beskikbaar by <https://www.takealot.com/rode-dead-cat-windshield-for-videomic-pro/PLID40364097> teen R595 (BTW ingesluit). Een per mikrofoon word benodig. Die namaakpels wat vir hierdie projek gebruik is, is egter gratis afvalstukke wat van 'n gawe naaldwerkster bekom is.



Figuur 4.9: Gelyktydige opnames met twee ICS-40300-mikrofone, sonder (bo) en met (onder) windbeskerming.



Figuur 4.10: Voorbeelde van ruimtelike oorvleueling as gevolg van mikrofoonspasiëring. Die skakering van die agtergrond dui lugdruk (klankgolwe) aan wat diagonaal oor die vertoonde area beweeg. As die frekwensie van die klankgolwe gelyk is aan $\frac{v}{d \sin(\alpha)}$ (links), meet beide mikrofone dieselfde sein – ’n geval wat presies ooreenstem met $\alpha = 0$. Oorvleueling het dus plaasgevind. Die werklike oorvleueling begin egter al by ’n halwe golflengte, oftewel ’n frekwensie van $\frac{v}{2d \sin(\alpha)}$ (regs).

Die ontstaan van so ’n oorvleueling is duideliker as daar op die kruiskorrelasie van ’n nou-band sein gelet word: Beskou ’n 3 Hz sinusgolf $s_1 = \sin(6\pi t)$ wat deur twee verskillende mikrofone gemeet word. Die eerste mikrofoon meet s_1 sonder vertraging. Die tweede mikrofoon meet sein s_2 , wat ’n tyd-vertraagde weergawe van s_1 is. Figuur 4.11 toon hierdie sinusgolwe (links) en die kruiskorrelasie tussen s_1 en s_2 (regs) vir verskillende tydvertraginge (rye). Die derde ry dui ’n vertraging van ’n halwe periode aan, en dit is duidelik dat daar nie uit die kruiskorrelasie bepaal kan word of die vertraging positief of negatief is nie. Op ’n soortgelyke manier oorvleuel die voorbeelde in rye 4 en 5 onderskeidelik met rye 2 en 1.

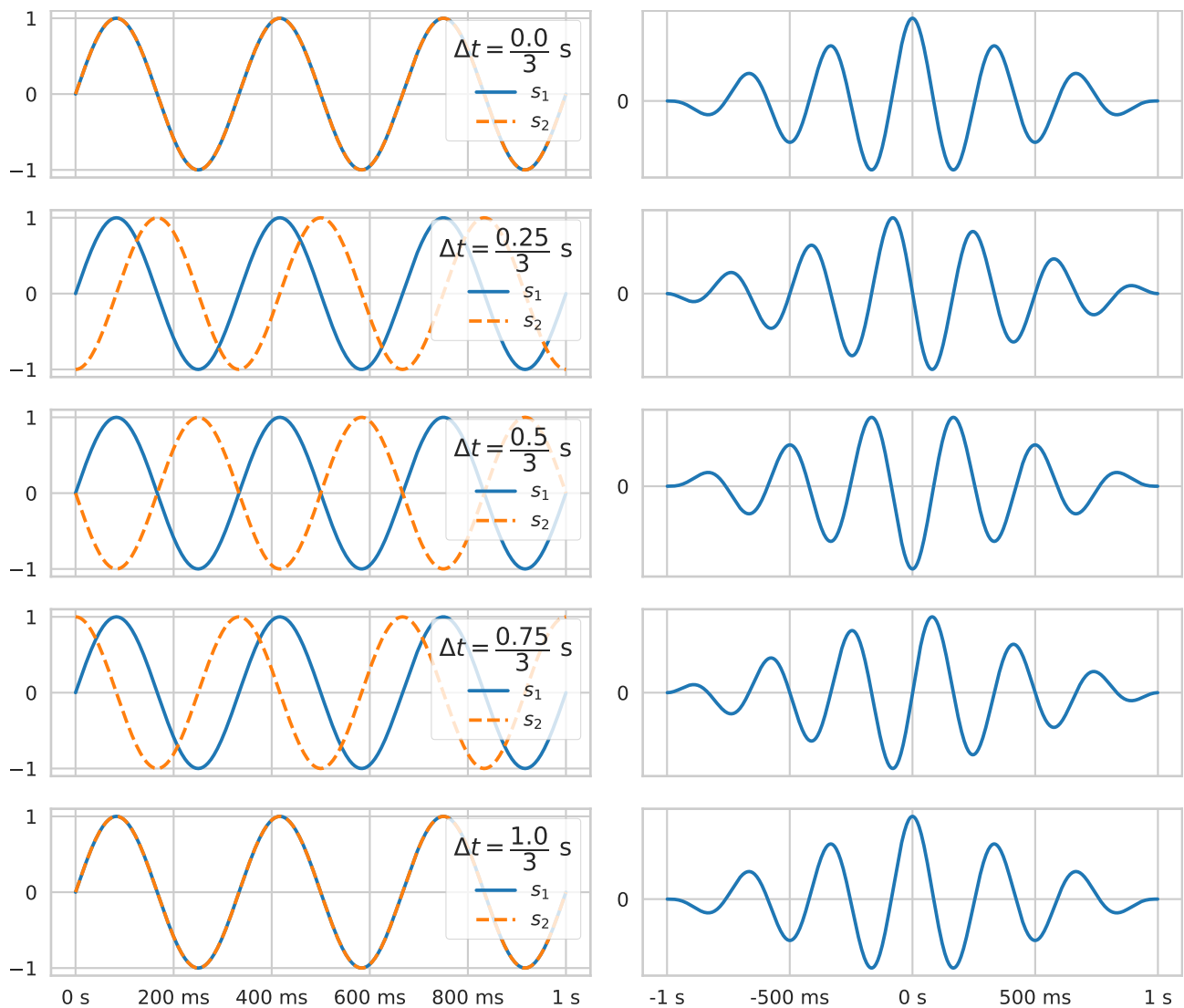
Vir ’n breë-band sein bied die kruiskorrelasie egter beter resultate. Namate meerdere frekwensies onder 3 Hz in die vorige voorbeeld bygevoeg word, begin die halfgolflengte-oorvleueling verdwyn. Figuur 4.12 toon die geval waar $s_1 = \sum_{f \in F} (\sin(2\pi ft))$ waar F tien eweredig gespasieerde frekwensies tussen 1 Hz en 3 Hz voorstel. Dit is duidelik uit hierdie figuur dat geen noemenswaardige oorvleueling plaasgevind het nie.

Olifantdreungeluide het harmoniese inhoud tot en met ongeveer 500 Hz. Uit Figuur 4.10 is dit duidelik dat die oorvleueling die ergste is wanneer $\alpha = 90^\circ$. Om alle oorvleueling te voorkom en goeie kruiskorrelasieresultate vir enige tipe sein te waarborg, is daar besluit om die afstand tussen die mikrofone met Vergelyking 4.2 te beperk:

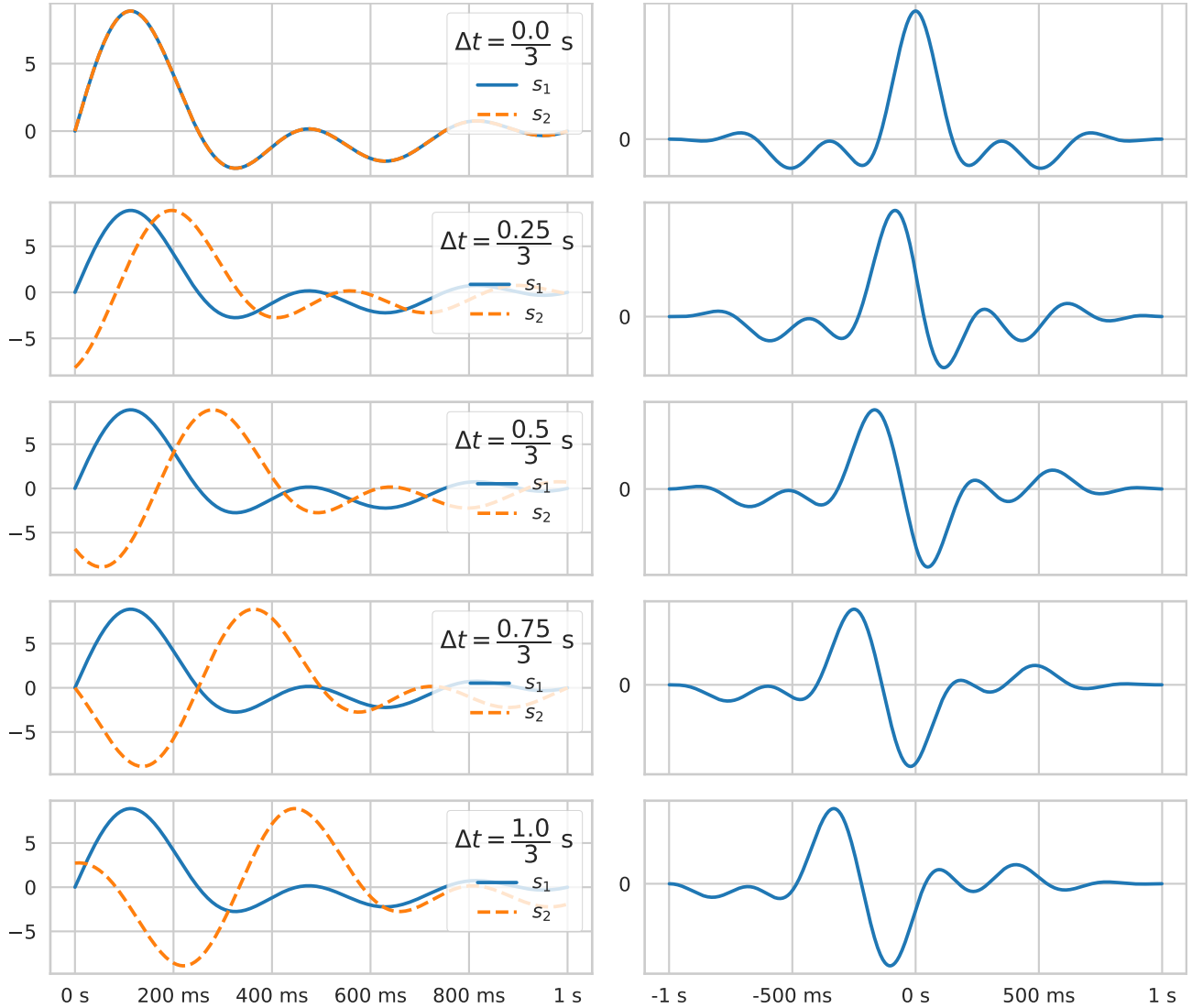
$$d < \frac{v}{2f_{\max} \sin(\alpha)} = \frac{343 \text{ m} \cdot \text{s}^{-1}}{2 \times 500 \text{ Hz} \times \sin(90^\circ)} = 34,3 \text{ cm} \quad (4.2)$$

Hoe verder die mikrofone uitmekaar is, hoe beter is die resolusie van die mikrofoonpaar. Dit is duidelik uit Vergelyking 4.1: Hoe groter d is, hoe minder sensitief is die beraamde rigting α vir foute in die beraming van die tydverskil τ .

’n Uiteindelijke mikrofoonafstand van 20 cm word gekies, en 6 mikrofone word in ’n sirkel met radius 20 cm gerangskik. Die montering van die mikrofone op ’n onderstel word in §9.3 bespreek.



Figuur 4.11: Links: 3 Hz sinusgolf $s_1 = \sin(6\pi t)$ en 'n vertraagde weergawe daarvan s_2 . Regs: Die kruiskorrelasie van s_1 en s_2 . Van bo na onder: Verskeie waardes vir die tydvertraging. Vir rye 3 tot 4 vind oorvleueling in die kruiskorrelasieresultaat plaas omdat die werklike vertraging nie daaruit teruggevind kan word nie.



Figuur 4.12: Links: Breë-band sein $s_1 = \sum_{f \in F} (\sin(2\pi ft))$ en 'n vertraagde weergawe daarvan s_2 . Regs: Die kruiskorrelasie van s_1 en s_2 . Van bo na onder: Verskeie waardes vir die tydvertraging. In teenstelling met Figuur 4.11 is hier oënskynlik geen oorveluling nie – 'n duidelike piek in die kruiskorrelasieresultaat is telkens by $-\Delta t$ sigbaar.

Hoofstuk 5

Oudiokoppelvlak

Selfs al word 'n mikrofoon met gepaste frekwensieweergawe aan 'n opnemer gekoppel, kan die lae frekwensies deur die opnemer afgesny word. Ondersoek na beskikbare klankopnemers is reeds in §2.1 ingestel, en daar is bevind dat daar nie geskikte beskikbare klankopnemers is nie.

Hierdie hoofstuk gee aandag aan die ontwerp van 'n eie oudiokoppelvlak wat by die ICS-40300-mikrofone pas. Die oudiokoppelvlak bestaan uit 'n voorversterker, 'n laagdeurlaatfilter om oorvleueling te voorkom en 'n versyferaar. Buiten die frekwensieweergawe moet die oudiokoppelvlak die volgende eienskappe hê, om goed met die mikrofone saam te werk:

- 'n Geskikte **intree**-impedansie – hoër is gewoonlik beter.
- Geskikte aanwins, sodat die maksimale omvang van die versyferaar gebruik word sonder om die sein af te sny.
- Lae intrinsieke ruis – verkieslik laer as dié van die mikrofoon.

5.1 Versyferaar

Hoe minder ruis daar in die stelsel is en hoe meer resolusie die versyferaar het, des te beter is die inherente **SNR** van die stelsel. 'n Stelsel met 'n hoër **SNR** sal geluide oor 'n langer afstand kan bespeur, want die amplitude van die sein verswak met afstand, terwyl die inherente ruis in die stelsel konstant bly. Vir hierdie rede is versyferaar met 24-bis monster-resolusie en lae ruis gesoek. Die **monstertempo** moet hoër as 2 kHz wees, as die **teen-oorvleuelingsfilter** by 1 kHz afsny.

Die MCP3903 versyferaar van Microchip het uitgestaan as 'n goedkoop¹ oplossing wat aan die bogenoemde spesifikasies voldoen. Hierdie versyferaar het 6 hoë-impedansie **intree**poorte wat gelyktydig gemonster word. Elke monster is 24 bisse (3 B) per kanaal, en die monstertempo kan tot by 'n maksimum van 64 kHz gestel word. 'n **SPI**-koppelvlak word voorsien om data aan 'n mikrobeheerder oor te dra. Geen ander versyferaar is ernstig oorweeg nie.

Die stroombaanuitleg van Microchip se MCP3903-ontwikkelingsbordjie ([Microchip, 2011a](#), pp. 16–21) is gebruik as 'n beginpunt vir die ontwerp van 'n eerste prototipe van 'n eie versyferaar-stroombaan. Die uiteindelijke stroombaan word in Figuur 5.1 getoon.

Die versyferaar het 11 24-bis registers wat deur **SPI** gelees of geskryf kan word. Registers 0x00–0x05 bevat die versyferde data van die 6 afsonderlike kanale, en registers 0x06–0x0A bevat konfigurasiedata en inligting oor die toestand van die versyferaar. Tabel 5.1 gee 'n opsomming van die relevante konfigurasie-registers en hoe dit vir hierdie projek opgestel moet word. Registers en bisse wat nie gelys word nie, word geïgnoreer, of by hul verstekwaarde gehou. 'n Noukeuriger beskrywing van die doel van elke register is beskikbaar in die datablad ([Microchip, 2011b](#)).

¹ R79 elk, by RS Online, ten tyde van skrywe. **BTW** ingesluit.

Tabel 5.1: Konfiguratie-registers van die MCP3903-versyferaar. Elke register is 24 bisse breed. Bis 23 is die mees beduidende bis in elke register.

Die versyferaar dien as slaaf op die **SPI**-bus. Die registers word dus deur die mikroverwerker wat as **SPI**-meester dien, opgestel (sien §6.5).

Elkeen van die 6 differensiële **intreepoorte** van die versyferaar het 'n **programmeerbare voorversterker (PGA, Programmable Gain Amplifier)**. Die aanwys van die voorversterkers word met behulp van die **GAIN**-register na 'n minimum van 0 dB gestel, om die toelaatbare **insetspanningsomvang** te maksimeer. Die versyferaar sny dan die sein af as die differensiële **insetspanning** $\pm 0,5$ V oorskry. Volgens die datablad (**Microchip, 2011b**) word akkuraatheid slegs gewaarborg indien die absolute **insetspanning** binne ± 1 V bly.

Die **STATUS|COM**-register word gebruik om die monster-resolusie na 24 bisse per monster te stel. Die **uittreepenne nDRA, nDRB en nDRC** word gestel om sinkroon te werk, sodat dit net nodig is om een van die drie penne aan die mikroverwerker te koppel. Hierdie lyne word laaggetrek wanneer die versyferde data gereed is. Sodra die volgende monster gereed is, word dit bo-oor die ou data in die kanaalregisters van die versyferaar geskryf. Die mikroverwerker moet dus tydig op veranderinge van die **nDRA**-lyn reageer, sodat monsters nie verlore gaan nie. Die werking hiervan word in §6.8 bespreek.

As die **intreesein** swak is, word afrondingsfoute in die versyferingsproses 'n dominerende bron van ruis. Om hierdie probleem teen te werk, bevat die versyferaar 'n **ruisverspreidings**-algoritme. So 'n algoritme versprei die ruis in die frekwensiedomein deur die afrondingsfoute stogasties te maak, eerder as deterministies. Die totale harmoniese vervorming word dus verbeter, terwyl die algehele **SNR** effens verswak word. Die presiese algoritme is die intellektuele eiendom van **Microchip** en word nie in die datablad bespreek nie. Hierdie algoritme word in die **CONFIG**-register aangeskakel.

Volgens die datablad moet die **mate van oor-monstering (OSR, Over-Sample Rate)** na die maksimumwaarde van 256 maal gestel word om 'n 24-bis resolusie te bereik (**Microchip, 2011b**, Tabel 5-2). Die waarde van **OSR**, saam met die skalering van die analoogklok, wat ook in die **CONFIG**-register gestel word, bepaal die **monstertempo** van die versyferaar. Die effektiewe **monstertempo** word met behulp van Vergelyking 5.1a bereken. Die vervaardiger spesifiseer dat die kristal wat gebruik word teen hoogstens 16,384 MHz moet ossileer (Vergelyking 5.1b) en dat die interne analoogklok na skalering hoogstens 5 MHz moet wees² (Vergelyking 5.1c). Verder word die kristal se weerstand deur Vergelyking 5.1d beperk.

$$f_S = \frac{f_X}{4 \times \text{OSR} \times \text{SKALERING}} \quad (5.1a)$$

$$f_X < 16,384 \text{ MHz} \quad (5.1b)$$

$$f_A = \frac{f_X}{\text{SKALERING}} < 5 \text{ MHz} \quad (5.1c)$$

$$R_M < 1,6 \times 10^6 \times \left(\frac{1}{f_X \times C_{\text{las}}} \right)^2 \quad (5.1d)$$

waar f_S = effektiewe monstertempo, in Hz,

f_X = kristalfrekwensie, in MHz,

OSR = mate van oormonstering in die versyferaar,

SKALERING = die interne klokskalering,

C_{las} = las-kapasitansie plus kapasitansie van die bord, in pF,

en R_M = weerstand van die kristal, in Ω .

Met 'n 4,9152 MHz-kristal, 256-voudige oormonstering en 'n klokskalering van 1 is die **monstertempo** $f_S = 4800$ Hz. Die kristal wat gekies is vir die toepassing is die **QCP9**, wat 'n weerstand

² Daar is 'n opsie om die stroomverbruik van die versyferaar te verhoog sodat die analoogklok tot by 8,192 MHz kan funksioneer, maar dit was nie nodig nie.

van hoogstens $150\ \Omega$ het (QANTEK Technology Corporation, datum onbekend). Die aanbevole laskapasitansie volgens die kristal se datablad is $10\text{--}32\ \text{pF}$. Die $18\ \text{pF}$ laskapasitansie in Figuur 5.1 voldoen dus aan die aanbeveling van die kristal-datablad, sowel as die beperking in Vergelyking 5.1d.

Die frekwensiegedrag van die versyferaar is tipies plat tot by $0\ \text{Hz}$ (Microchip, 2011b, Figuur 5-2).

5.2 Voorversterker

In 'n soortgelyke projek het Duckitt (2008, bl. 49–51) die PGA2500 voorversterker van TI aanbeveel. Hierdie skyfie bied verskeie voordele:

- Volledig differensiële intree- en uittreepoorte.
- Lae ruis en min harmoniese vervorming.
- Digitaal verstelbare aanwins van $10\ \text{dB}$ tot $65\ \text{dB}$ in spronge van $1\ \text{dB}$.
- 'n uittreepen wat aandui wanneer die intreesein te groot is.

Daar is egter ook 'n paar bepalende nadele, wat die PGA2500 ongeskik maak vir hierdie projek:

- Positiewe én negatiewe $5\ \text{V}$ -toevoerspannings word benodig. Dit verhoog die kompleksiteit van die ontwerp.
- Die komponent is betreklik duur (ongeveer $\$18$ per kanaal by DigiKey ten tyde van skrywe).
- Die uittree swaai rondom $0\ \text{V}$, terwyl baie versyferaar slegs positiewe spannings kan meet.

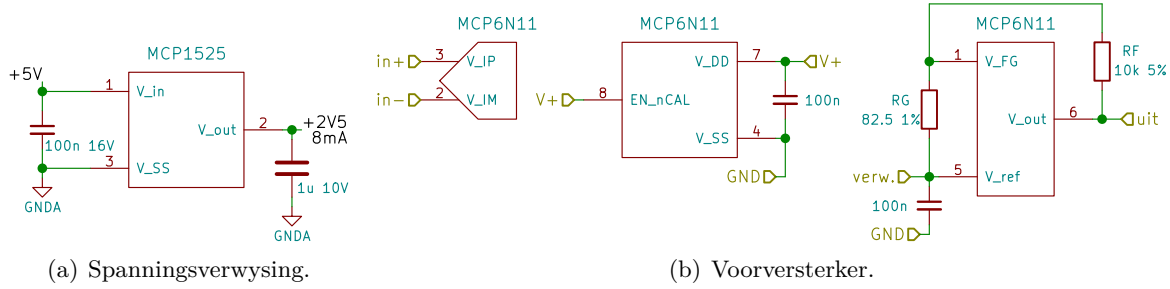
'n Heelwat goedkoper voorversterker is gevind. Die MCP6N11-100 van Microchip³ het verskeie voordele bó die PGA2500:

- 'n Enkele $5\ \text{V}$ -toevoerspanning word benodig.
- 'n Lyn-na-lyn ($-0,2\text{--}5,15\ \text{V}$) intreepanspanning word aanvaar.
- Stroomverbruik is laag: ongeveer $800\ \mu\text{A}$ per kanaal, teenoor $30\ \text{mA}$ per bron per kanaal vir die PGA2500.
- Die uittree swaai rondom 'n verwysingspanning van die ontwerper se keuse (byvoorbeeld $2,5\ \text{V}$, wat ideaal is vir die intree van die versyferaar).
- Die komponent is goedkoop: $\text{R}25,42$ elk by RS Online ten tyde van skrywe, teenoor $\text{R}315,02$ vir die PGA2500 (BTW ingesluit).

Die MCP1525 van Microchip (Figuur 5.2(a)) word gebruik om 'n $2,5\ \text{V}$ spanningsverwysing te skep vir die V_{ref} -intree van die voorversterker. Figuur 5.2(b) toon die stroombaandiagram van een voorversterker-kanaal.

Die uittreepanspanning van die voorversterker word deur Vergelykings 5.2 benader (Microchip, 2011d, bl. 9). Die weerstandwaardes R_F en R_G word in Figuur 5.2(b) gegee.

³ Verskeie modelle van die MCP6N11-skyfie bestaan. Die MCP6N11-100 het laer ruis, beter temperatuurhantering, beter kragbronverwerping, beter gemene-modus intreeverwerping en 'n hoër eenheidsaanwinsbandwydte as al die ander modelle. Dit moet egter in 'n konfigurasie met 'n minimum geslote-lus aanwins van 100 gebruik word om stabiel te wees.



Figuur 5.2: Voorversterker met behulp van Microchip MCP6N11 en 2,5 V-spanningsverwysing met behulp van Microchip MCP1525. V+ is aan +5V gekoppel, en verw. aan +2V5_8mA.

$$G_{DM} = 1 + \frac{R_F}{R_G} \quad (5.2a)$$

$$V_{out} = V_{ref} + G_{DM} (1 + g_E) (V_{DM} + V_E) \quad (5.2b)$$

waar G_{DM} = differensiële aanwinst,

V_{out} = uittreespanning,

V_{ref} = verwysingspanning,

g_E = differensiële aanwinstfout

V_{DM} = differensiële intreespanning,

en V_E = totale GS-spanningsfout by die intree.

Die minimum en maksimum differensiële aanwinst word in Vergelyking 5.3 bereken. Hierdie speling bestaan a.g.v. die onakkuraatheid van weerstandwaardes. Die speling is aanvaarbaar, omdat die vermenigvuldiging van 'n sein met 'n konstante nie die posisie van die piek van die kruiskorrelasie met 'n ander sein beïnvloed nie, en daarom sal dit ook nie die klankbronrigtingberaming affekteer nie.

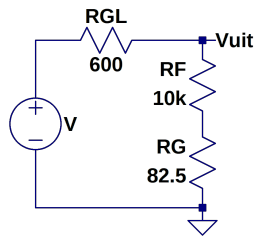
$$\begin{aligned} 1 + \frac{R_{F,min}}{R_{G,maks}} < G_{DM} < 1 + \frac{R_{F,maks}}{R_{G,min}} \\ \therefore 1 + \frac{0.95 \times 10 \text{ k}\Omega}{1.01 \times 82,5 \Omega} < G_{DM} < 1 + \frac{1.05 \times 10 \text{ k}\Omega}{0.99 \times 82,5 \Omega} \\ \therefore 115 < G_{DM} < 130 \end{aligned} \quad (5.3)$$

Laastens word die Thévenin-ekwivalente uittreeweerstand van die voorversterkerbaan ook bereken, sodat die filter aangepas kan word om daarvoor te kompenseer. Die oop-lus uittreeweerstand van die MCP6N11-100 is tipies 600Ω volgens die datablad⁴. Onder die aanname dat die stroom wat by penne V_{FG} en V_{ref} (Figuur 5.2(b)) invloei weglaatbaar klein is, kan die uittree van die voorversterker met die stroombaan in Figuur 5.3 voorgestel word. Die Thévenin-ekwivalente uittreeweerstand en uittreespanning word dus deur Vergelykings 5.4 gegee.

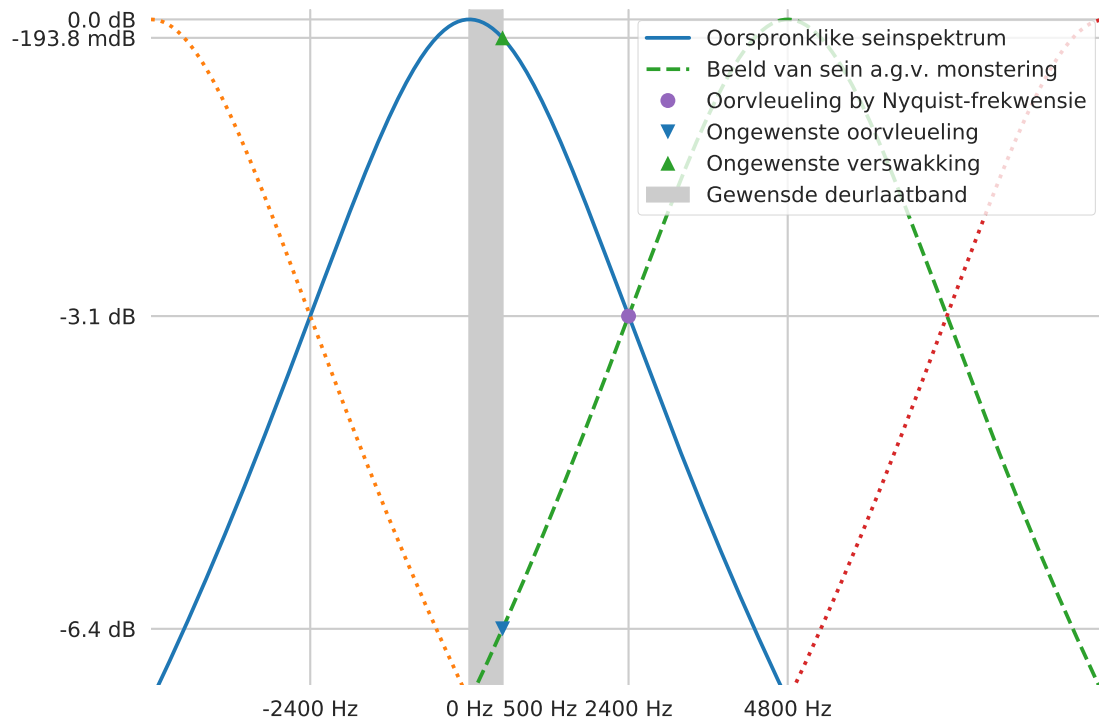
$$V_{Th} = \frac{V (R_F + R_G)}{R_{GL} + R_F + R_G} = 0,9438V \quad (5.4a)$$

$$R_{Th} = R_{GL} \parallel (R_F + R_G) = \frac{R_{GL} (R_F + R_G)}{R_{GL} + R_F + R_G} = 566,3 \Omega \quad (5.4b)$$

⁴ Die datablad (Microchip, 2011d, Figuur 2-36) toon dat die geslote-lus uittreeweerstand ook 'n funksie van frekwensie sal wees. Daar word egter gewerk met die aanname dat die uittreeweerstand 'n konstante is, om die ontwerp te vereenvoudig. Die gedrag van die baan word in §5.5 getoets om te verifieer of die ontwerp-aannames aanvaarbaar is.



Figuur 5.3: Ekwivalente **uittreebaan** van die voorversterker, waarmee die Thévenin-ekwivalente **uittree** bereken word.



Figuur 5.4: Oorvleueling in die frekwensiedomein as gevolg van monsterring, wanneer 'n eerste orde passiewe laagdeurlaatfilter ($R = 1 \text{ k}\Omega$ en $C = 68 \text{ nF}$) gebruik word. Die mate van oorvleueling tussen 500 Hz en die Nyquist-frekwensie is nie van belang nie, omdat dit ná monsterring digitaal gefilter kan word. Die mate van oorvleueling in die gewenste **deurlaatband** 0–500 Hz is egter wel van belang, siende dat dit nie na monsterring verwyder kan word nie.

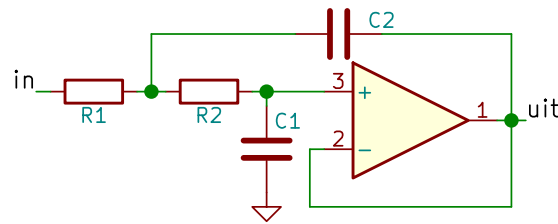
5.3 Teen-oorvleuelingsfilter

Met 'n **monstertempo** van 4800 Hz is die Nyquist-frekwensie by 2400 Hz. Die harmonieke van die olifandreungeluide strek gewoonlik tot by hoogstens 500 Hz. 'n **Teen-oorvleuelingsfilter** moet dus ontwerp word met minimale seinverswakking by 500 Hz en maksimale verswakking by 4300 Hz (500 Hz onder die **monstertempo**).

Die eerste stap is om te bepaal of 'n passiewe, eerste-orde laagdeurlaatfilter die oorvleueling voldoende sal beperk. Figuur 5.4 toon die grootte-weergawe van so 'n filter, met $R = 1 \text{ k}\Omega$ en $C = 68 \text{ nF}$. Die maksimum oorvleueling in die gewenste **deurlaatband** is $-6,4 \text{ dB}$, wat te veel is.

'n Hoër-orde passiewe filter sal 'n nouer oorgang hê tussen die **deurlaatband** en die **keerband**, maar die **intree**-impedansie van die baan word laer met elke orde wat bygevoeg word. Vir hierdie rede is daar besluit om 'n aktiewe laagdeurlaatfilter te ontwerp. 'n Aktiewe filter kan 'n baie nouer **oorgangsband** en terselfdetyd 'n hoër **intree**-impedansie hê, omdat dit van operasionele versterkers gebruik maak.

Oor die algemeen word aktiewe laagdeurlaatfilters gerealiseer deur verskeie tweede orde filters aanmekaar te skakel. Figuur 5.5 toon die Sallen-Key-topologie wat gewoonlik gebruik word.



Figuur 5.5: Aktiewe laagdeurlaat-opset met Sallen-Key-topologie en eenheids-aanwins.

Vergelyking 5.5 toon die teoretiese frekwensieweergawe van hierdie filter as dit met ideale komponente gebou word. 'n Toepassingsnota van [Texas Instruments \(2002\)](#) bied 'n meer gedetailleerde ontleding van die Sallen-Key-topologie vir hoog- en laagdeurlaatfilters.

$$H(f) = \frac{f_p^2}{-f^2 + \frac{jff_p}{Q} + f_p^2}$$

$$\text{waar } f_p = \frac{1}{2\pi\sqrt{R_1R_2C_1C_2}}, \quad (5.5)$$

$$\text{en } Q = \frac{\sqrt{R_1R_2C_1C_2}}{R_1C_1 + R_2C_1}$$

By die pool-frekwensie f_p is die frekwensieweergawe $H(f_p) = \frac{Q}{j}$. Dit beteken dat die aanwins by die poolfrekwensie Q is, met 'n faseskuif van 90° .

Q word ook die kwaliteit-faktor van die filter genoem. Sallen-Key filters met lae Q -waardes staan bekend as Bessel-filters. Hierdie filters het 'n oorgedempte trapweergawe en 'n breë **oorgangsband**. Chebychev-filters het hoë Q -waardes. Hierdie filters het 'n ondergedempte trapweergawe en 'n nou **oorgangsband**. Tussenin is Butterworth-filters, wat 'n kritiese gedempte trapweergawe het.

Chebychev-filters vertoon rimpels in die **deurlaatband**, terwyl Bessel-filters reeds verswakking in die **deurlaatband** begin toon. Butterworth-filters is maksimaal plat in die **deurlaatband**.

Figuur 5.6 toon die effek van Q en f_p op die frekwensiegedrag $H(f)$. Meer filterstadia (oftewel 'n hoër-orde filter) is dus nodig om dieselfde **oorgangsband** te kry met 'n Bessel-filter as met 'n Chebychev-filter.

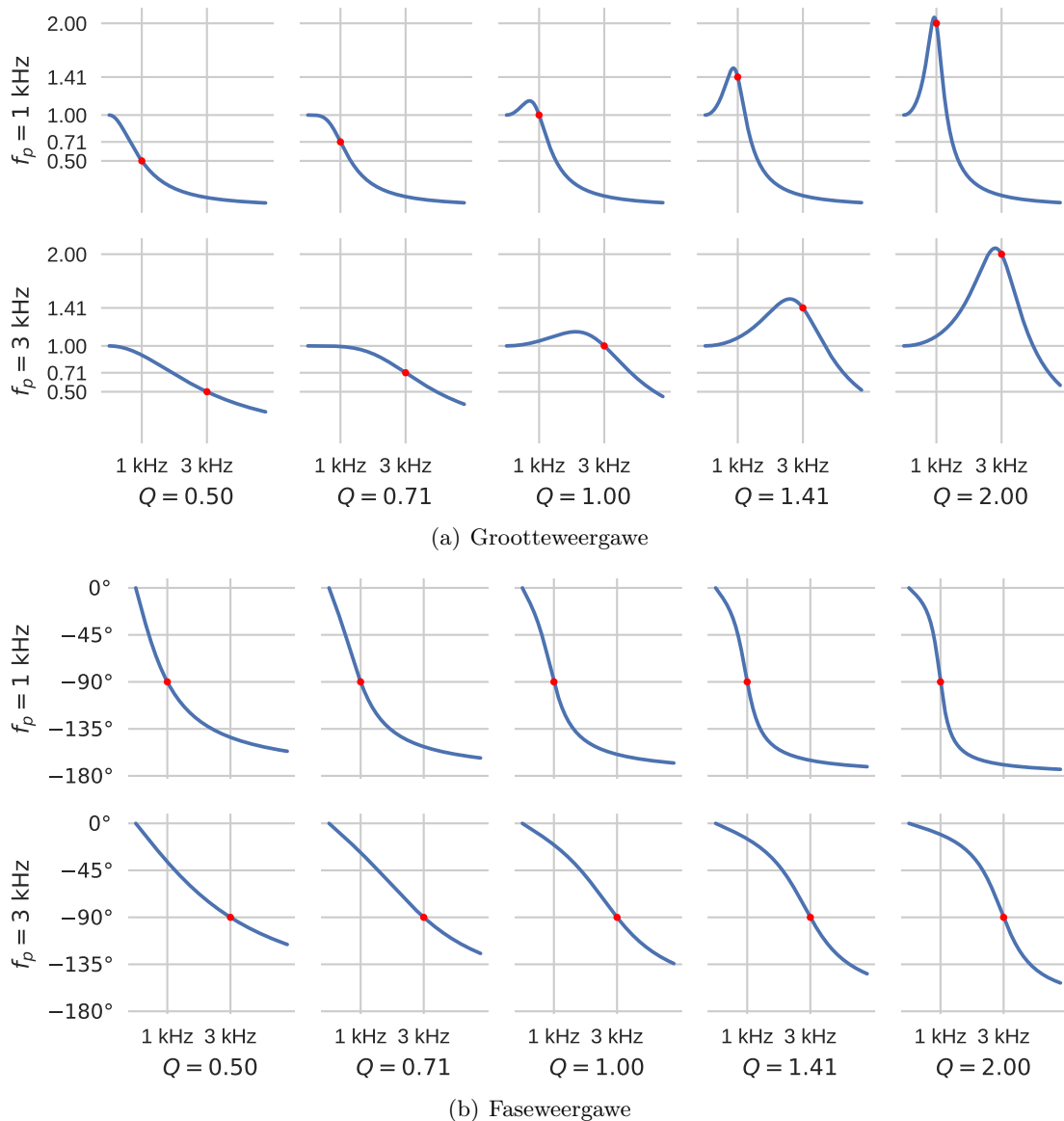
Meer gevorderde aktiewe filters bestaan ook: Inverse Chebychev-filters en elliptiese filters voeg 'n zero by en kan nog nouer **oorgangsbande** bied. Die stroombane is egter meer kompleks en word daarom nie verder oorweeg nie.

Daar is besluit op 'n agste-orde Chebychev-filter, wat met 'n viervoudige operasionele versterker geïmplementeer kan word. Die skerp afsny van die Chebychev-opset is ideaal vir 'n **teen-oortreuefilter**. Mits die grootte weergawe en fase weergawe van die filter by elke kanaal van die versyferaar dieselfde is, is die rimpels in die **deurlaatband** ook nie 'n probleem nie.

Die volgorde van die filterstadia is ook van belang. Sommige stadia het hoë Q -waardes en ander laer. As die hoë Q -waardes eerste geplaas word, vind meer versterking vroeër plaas, wat goed is vir ruisverwerping. Die gevaar is egter dat die sein afgesny word deur die beperkte **uittreeswaai** van die operasionele versterker. Om te optimeer vir maksimale spanningsomvang, word die stadia met lae Q -waardes dus eerste geplaas.

Die aanwins in die **deurlaatband** (wat 0 Hz insluit) moet 0 dB wees, om verskeie redes:

- Die **inset**spanning sal 'n **GS**-komponent van ongeveer 2,5 V hê, sodat die operasionele versterker vanaf 'n enkele 0–5 V bron kan werk. Hierdie waarde moet nie versterk of verswak word nie, want dit sal die sein uit die aanvaarbare spanningsomvang van die operasionele versterker skuif.



Figuur 5.6: Grootte- en faseweergawe van 'n Sallen-Key-laagdeurlaatfilter vir verskeie poolfrekwensies f_p en kwaliteit-faktore Q . f_p is die frekwensie waar die faseskuif 90° bereik, en Q is die aanwys by frekwensie f_p .

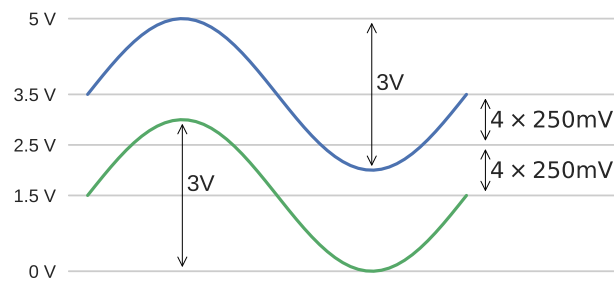
- Die **uittree** van elke operasionele versterker het 'n **GS**-spanningsfout, wat met die aanwys van daardie stadium vermenigvuldig word. Indien die aanwys groter is, moet meer voorsiening gemaak word vir seine wat moontlik kan skuif (sien Figuur 5.7).
- Die aanwys kan by die voorversterker verhoog word indien nodig (sien §5.2).

Die filter is gedeeltelik met behulp van **TI** se **WEBENCH**⁵ en **Analog Devices** se **Filter Wizard**⁶ ontwerp. Hierdie aanlyn-hulpmiddels laat die ontwerper toe om filter-spesifikasies in te voer en stel dan komponentwaardes voor om die ontwerp te realiseer. Spesifikasies wat ingevoer is, sluit in:

- $-0,1$ dB aanwys by 1 kHz. Die oorgang tussen die **deurlaatband** (0–1000 Hz) en die **keerband** (> 1000 Hz) begin dus hier. Dit skuif die ergste rimpel-gedrag van die Chebyshev-filter uit

⁵Beskikbaar by <http://webench.ti.com/>

⁶Beskikbaar by <http://www.analog.com/designtools/en/filterwizard/>



Figuur 5.7: Maksimum omvang van uitsetspannings met maksimum en minimum GS-spanningsfoute.

die 0–500 Hz-band, waar die meeste olifantgeluide voorkom, na die 500–1000 Hz-band, wat later digitaal uitgefilter kan word.

- –100 dB aanwys by 4300 kHz. Die frekwensies 4300–4800 Hz oorvleuel met die gewenste **deurlaatband** 0–500 Hz, en daarom is dit wenslik om dit maksimaal te verswak.
- Agste orde Chebychev-opset.
- Optimeer volgorde van filterstadia vir spanningsomvang 0–5 V (lae Q eerste), eerder as vir lae ruis (hoë Q eerste).

Omdat die Q en f_p sensitief is vir verandering in die waardes van R_1 , R_2 , C_1 en C_2 , moet kapasitortoleransies van 5 % of laer en weerstandtoleransies van 1 % of laer gebruik word. Alhoewel baie verskillende weerstand- en kapasitorwaardes die gewenste frekwensieweergawe sal laat realiseer, moet die volgende beperkings in ag geneem word:

- Groter weerstandwaardes vereis kleiner kapasitors en andersom.
- Baie klein kapasitors (onder 100 pF) kan veroorsaak dat die parasitiese kapasitansie van die stroombaanbord begin domineer, wat die frekwensiegedrag verander.
- Baie klein weerstandwaardes kan veroorsaak dat die weerstand van die bane op die stroombaanbord begin domineer, wat die frekwensiegedrag verander.
- Baie groot weerstandwaardes veroorsaak ruis.
- Baie groot kapasitors neem fisies meer plek op die stroombaanbord in beslag, is duurder en het swakker toleransies.

Kapasitors met hoë-kwaliteit diëlektrika, soos NP0, word verkies.

TI se LMV344 viervoudige operasionele versterker is gekies (Texas Instruments, 2012). Dit is geskik vir lae-drywing toepassings met 'n enkele 5 V-toevoerspanning. Die **intree** kan 'n gemode-modus spanning tot by 5 V hanteer en die **uitsetspanning** kan tot by die toevoerspannings swaai. Die **insetstroom** is baie klein (tipies 1 pA). Dit het 'n aanwys-bandwydte-produk van 1 MHz, wat baie hoër is as die frekwensie van seine wat hierdie filter sal deurlaat. Die komponent is betreklik goedkoop en maklik bekombaar⁷.

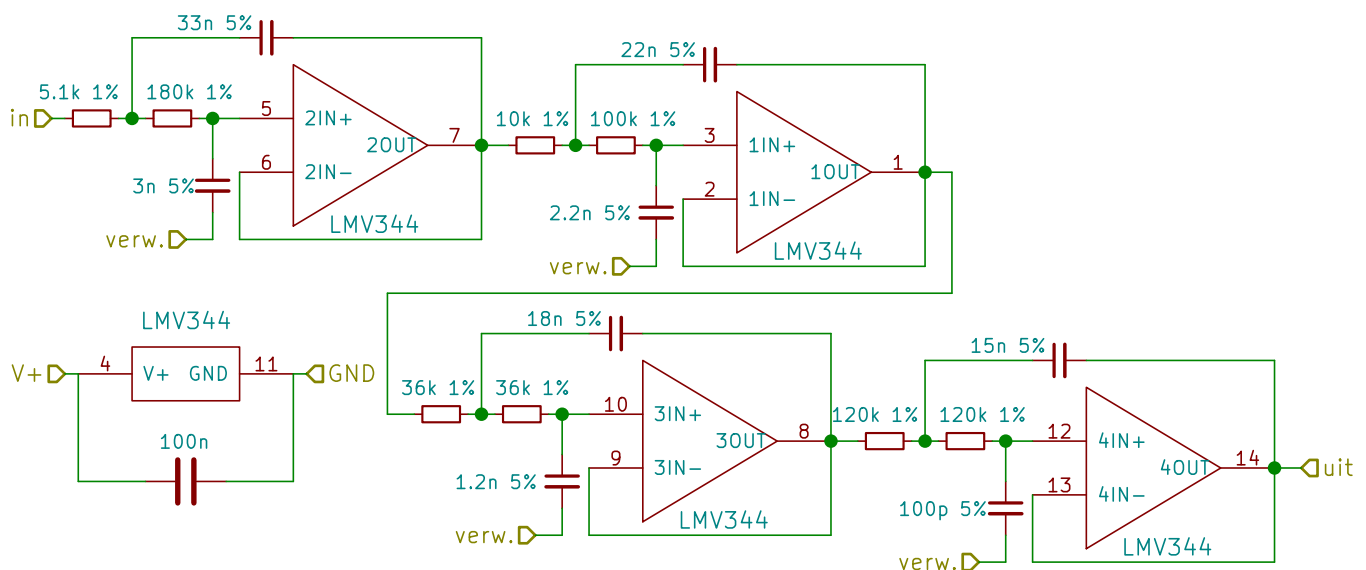
Die GS-spanningsfout by die **uittree** van die LMV344 is tipies ± 250 mV, wat beteken dat die **uitsetspanning** na 4 stadia in die ergste geval ± 1 V sal afwyk. Figuur 5.7 wys dat die **uitsetspanning** in hierdie geval nog steeds 3 V_{pp} kan swaai voordat dit geknip word.

Om vir die **uittree**weerstand van die voorversterker (sien §5.2) en die **intree**weerstand van die versyferaar (sien Vergelyking 5.4b) te kompenseer, is dit nodig om die komponentwaardes van die filter effens aan te pas. Tabel 5.2 gee die waardes wat deur die AD Filter Wizard voorgestel is, saam met die waardes wat in die finale stroombaandiagram in Figuur 5.8 gebruik is.

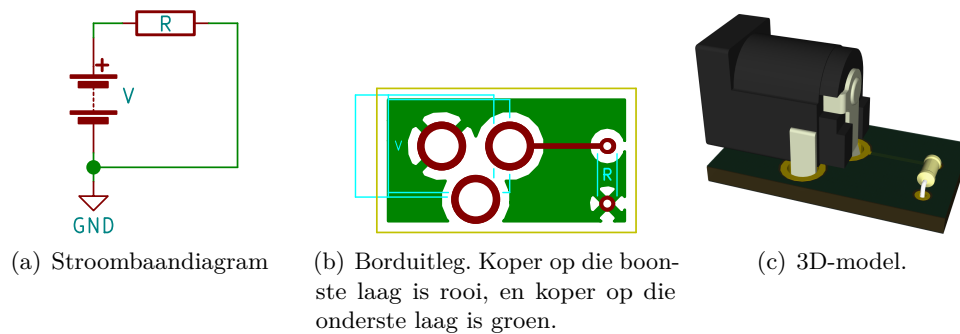
⁷R8,42 elk by RS Online. BTW ingesluit.

Stadium	Parameter	AD Filter Wizard	Finale ontwerp
1	R_1	5,62 k Ω	5,1 kΩ
	R_2	178 k Ω	180 kΩ
	C_1	3 nF	3 nF
	C_2	30 nF	33 nF
	f_p	530,42 Hz	527,94 Hz
	Q	0,544	0,543
2	R_1	10 k Ω	10 k Ω
	R_2	100 k Ω	100 k Ω
	C_1	2,2 nF	2,2 nF
	C_2	22 nF	22 nF
	f_p	723,43 Hz	723,43 Hz
	Q	0,909	0,909
3	R_1	35,7 k Ω	36 kΩ
	R_2	35,7 k Ω	36 kΩ
	C_1	1,3 nF	1,2 nF
	C_2	18 nF	18 nF
	f_p	921,6 Hz	951,24 Hz
	Q	1,86	1,94
4	R_1	115 k Ω	120 kΩ
	R_2	115 k Ω	120 kΩ
	C_1	110 pF	100 pF
	C_2	15 nF	15 nF
	f_p	1,08 kHz	1,08 kHz
	Q	5,84	6,12

Tabel 5.2: Die weerstand- en kapasitorwaardes van elke stadium van die teenoorvleuelingsfilter. Die waardes wat deur die AD Filter Wizard voorgestel is, moes effens aangepas word. In stadia 1, 3 en 4 is die waardes afgerond tot die naaste beskikbare komponentwaardes. Daarna is stadia 1 en 4 aangepas om onderskeidelik vir die **uittreeweerstand** van die voorversterker en die **intreeweerstand** van die versyferaar te kompenseer.



Figuur 5.8: Agste-orde Chebychev-laagdeurlaatfilter met behulp van TI LMV344 viervoudige operasionele versterker. Die **intreepoort** is direk aan die **uittreepoort** van die voorversterker in Figuur 5.2(b) gekoppel.



Figuur 5.9: 'n Eenvoudige stroombaan om die terugvloei pad van 'n stroom te demonstreer. As 'n battery by die batterysok ingeprop word, sal daar nie net 'n stroom deur die spoor op die boonste laag en deur die weerstand vloei nie, maar ook deur die grondlaag,

5.4 Vervaardiging

Die stroombaan het deur verskeie iterasies gegaan. In die vroegste weergawes is die voorversterker, filter en versyferaar op aparte bordjies geplaas. In die finale weergawe is al drie dele op een bordjie.

Die fisiese stroombaan is met KiCad uitgelê. Dit is 'n tydrowende, intuïtiewe optimeringsproses wat nie wiskundig bereken kan word nie. Die volgende beginsels is in ag geneem met die uitleg van die bord:

- Probeer om alle spore aan dieselfde kant van die bord te hou, sodat die ander kant van die bord as 'n grondvlak gebruik kan word. Dit beperk ruis en verhoog die stabiliteit van die baan.
- Hou alle komponente aan dieselfde kant van die bord om soldeerwerk makliker te maak.
- Hou spore so breed en kort as moontlik om weerstand te minimeer.
- Hou spore wat baie stroom dra weg van sensitiewe dele van die stroombaan, soos spore wat swak (onversterkte) seine dra, om te keer dat ongewenste spannings op daardie seine geïnduseer word (Faraday se wet).
- Let op die terugvloei pad van strome. Strome vloei altyd langs die pad van laagste impedansie. Vir **GS** is impedansie bloot 'n weerstand, wat deur die dikte en lengte van die geleiers bepaal word. **GS** sal dus langs die kortste moontlike pad terugvloei. Vir **WS** het die impedansie egter 'n induktiewe komponent wat deur die magneetveld beïnvloed word, en daarom vloei veranderende strome⁸ in 'n grondvlak oor die algemeen terug na die kragbron langs 'n pad wat direk onder die spoor is waarmee dit in die teenoorgestelde rigting gevloei het. Indien hierdie pad van laagste impedansie onderbreek word, moet die stroom langs 'n ander pad vloei en verhoog dit vanselfsprekend die impedansie. As so 'n terugvloei stroom langs sensitiewe seine beweeg, sal ongewenste elektromagnetiese induksie plaasvind.

'n Triviale voorbeeld word in Figuur 5.9(a) gegee. As 'n battery V aan die batterysok gekoppel word, sal 'n stroom $I = \frac{V}{R}$ deur die spoor op die boonste vlak en deur weerstand R vloei. Daardie stroom moet in die grondvlak na die negatiewe batteryterminaal terugvloei. Die grondvlak word egter onderbreek deur die monteringsgate van die batterysok, en die stroom moet met 'n ompad vloei. Dit is ekwivalent aan 'n ongewenste weerstand en induktor wat in serie met weerstand R geskakel word.

⁸ **GS** wat skielik aan- of afgeskakel word, moet natuurlik ook as **WS** beskou word.

Dit raak meer gekompliseerd wanneer komponente met meer as twee penne, soos operasionele versterkers of geïntegreerde stroombane, bygevoeg word, omdat dit nie altyd duidelik is hoeveel stroom by elke pen in-of uitvloei nie (Brokaw, 2000). So 'n komponent kan egter as 'n nodus beskou word waarvoor Kirchhoff se stroomwet steeds moet geld.

- Let op die posisie van **afvlakkapasitors**. Dit is nie genoeg om die positiewe pen van die **afvlakkapasitor** naby aan die operasionele versterker of geïntegreerde stroombaan te plaas nie. Verskillende punte op die stroombaanbord wat aan **GND** gekoppel is, het nie noodwendig dieselfde impedansie nie. Die negatiewe pen van 'n **afvlakkapasitor** moet dus so geplaas word om die induktansie (nie noodwendig die fisiese afstand nie) van die terugvloeiopad te minimeer (Brokaw, 2000).

Die finale uitleg is na PCBWay gestuur vir vervaardiging. Figuur 5.10 toon die resultaat. Alle komponente is met die hand gesoldeer.

5.5 Simulasies en metings

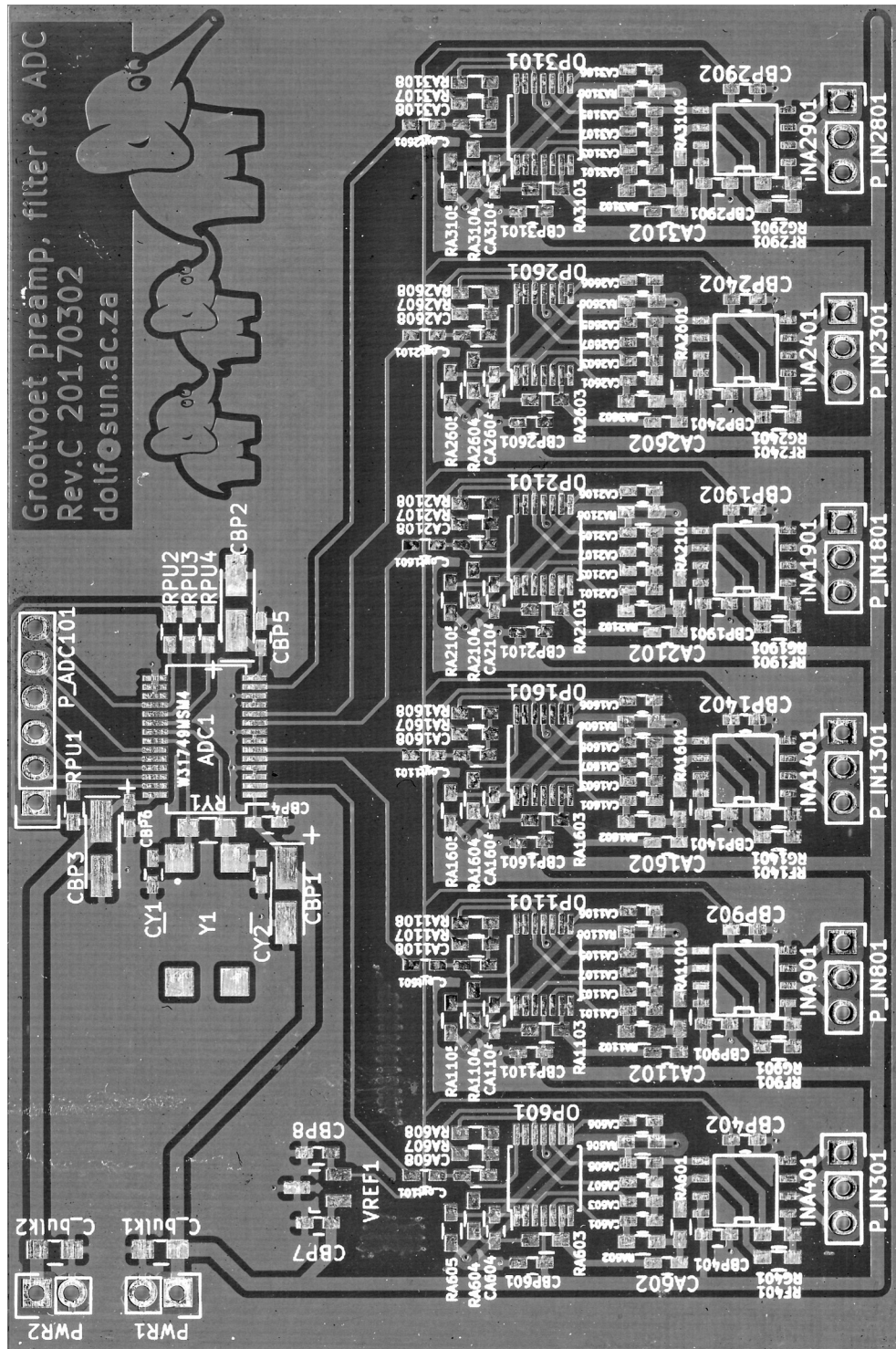
Voor vervaardiging is die voorversterker en filter met LTSpice gesimuleer. In hierdie simulasies word die voorversterker deur 'n afhanklike spanningsbron voorgestel, met weerstande om die **intree**- en **uittree**-impedansies te modelleer. Die Chebychev-filter is met ideale operasionele versterkers gesimuleer. Figuur 5.11 toon die opstelling in LTSpice.

Figuur 5.12 vergelyk die resultate van die simulase met gemete resultate van die baan wat gebou is. Binne die frekwensie-omvang van belang (8–500 Hz) stem die gemete en gesimuleerde resultate goed ooreen.

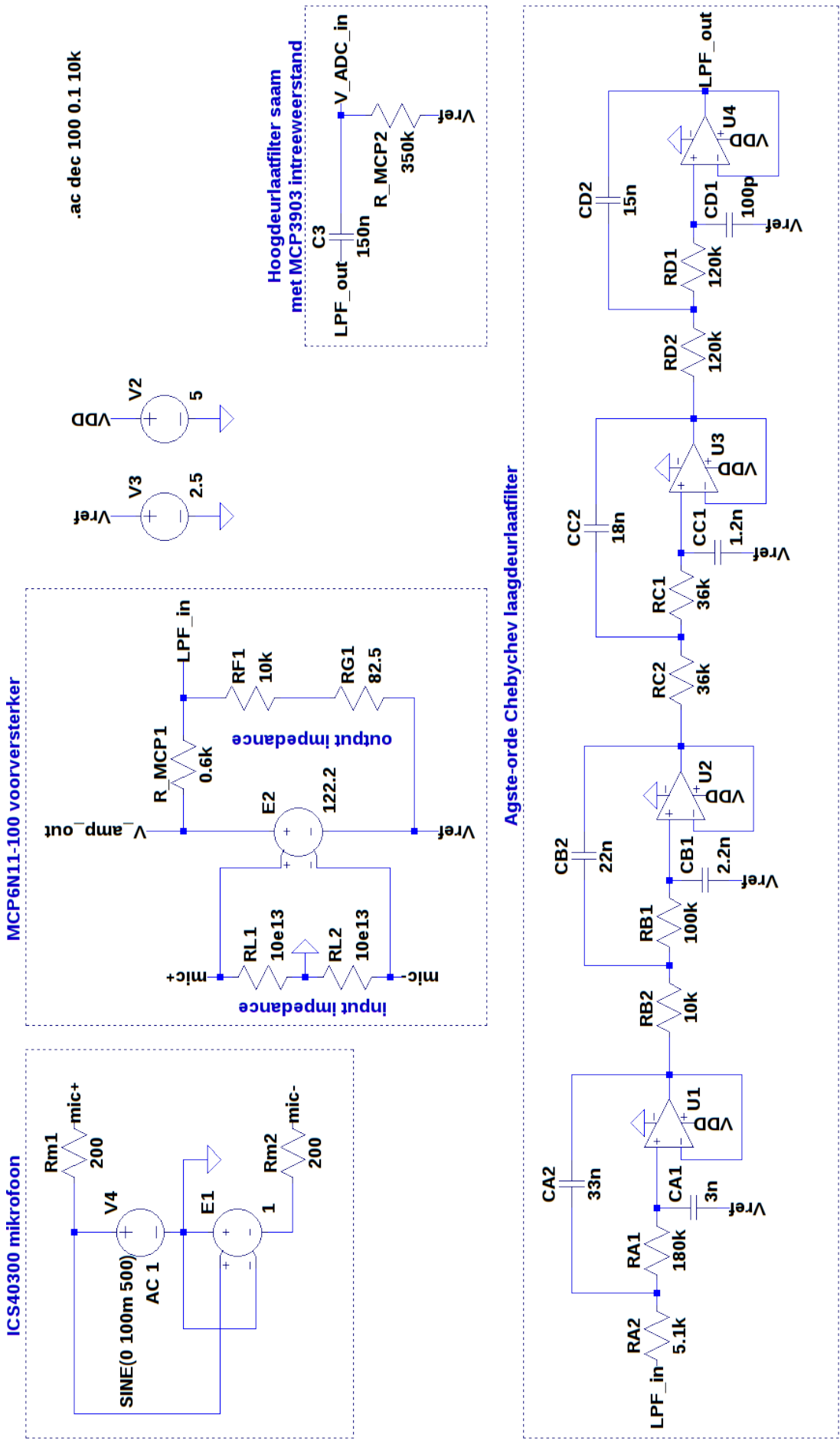
Die fase-inligting in die seine is belangrik vir klankbronopsoring. Indien die faseweergawe van al die kanale nie dieselfde is nie, sal dit die beraming van die klankbron se posisie omvergooi. Vyf van die filterkanale wat gebou is, is getoets en die standaardafwyking as 'n funksie van frekwensie word in Figuur 5.13 gegee.

Die afwyking van die fase in Figuur 5.13 lyk min, maar het verreikende gevolge vir die klankbronrigtingberaming. 'n Faseverskil van $0,4^\circ$ by 10 Hz stem ooreen met 'n tydvertraging van $\frac{0,4^\circ}{360^\circ \times 10 \text{ Hz}} \approx 111,1 \mu\text{s}$. Volgens Vergelyking 4.1 veroorsaak dit dus 'n hoekfout van $10,985^\circ$ in die rigtingberaming as die spoed van klank $343 \text{ m} \cdot \text{s}^{-1}$ is en die mikrofone 20 cm uitmekaar is. Dieselfde faseverskil by 300 Hz is egter slegs 'n $4,16 \mu\text{s}$ vertraging, wat ooreenstem met 'n hoekfout van $0,409^\circ$. Die tydvertraging word egter met behulp van kruiskorrelasies beraam, wat al die frekwensies in ag neem, en dus word 'n kleiner fout in praktyk verwag.

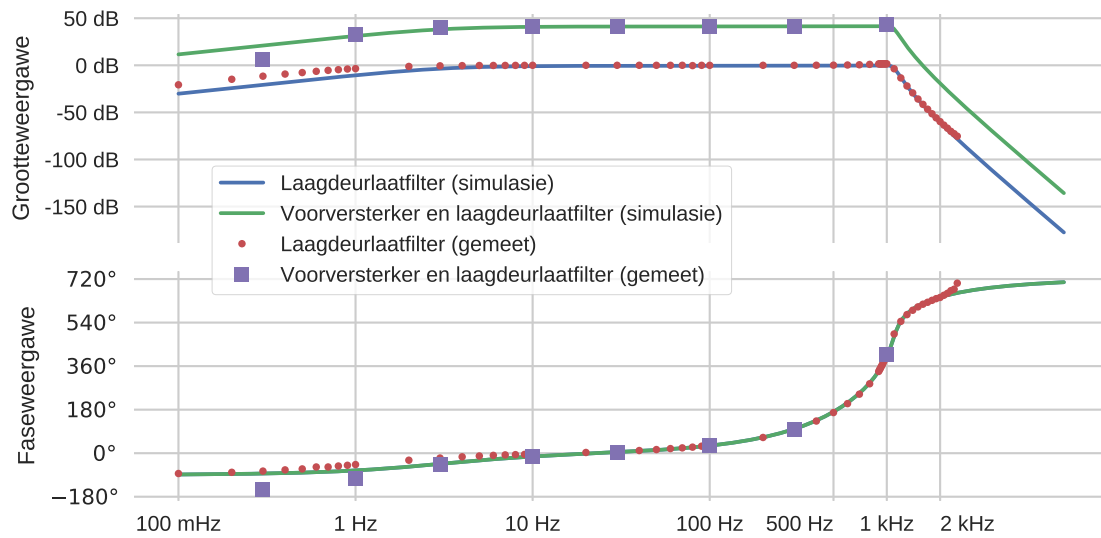
Toekomstige verbeterings aan hierdie stelsel kan die verbetering van hierdie fasegedrag insluit.



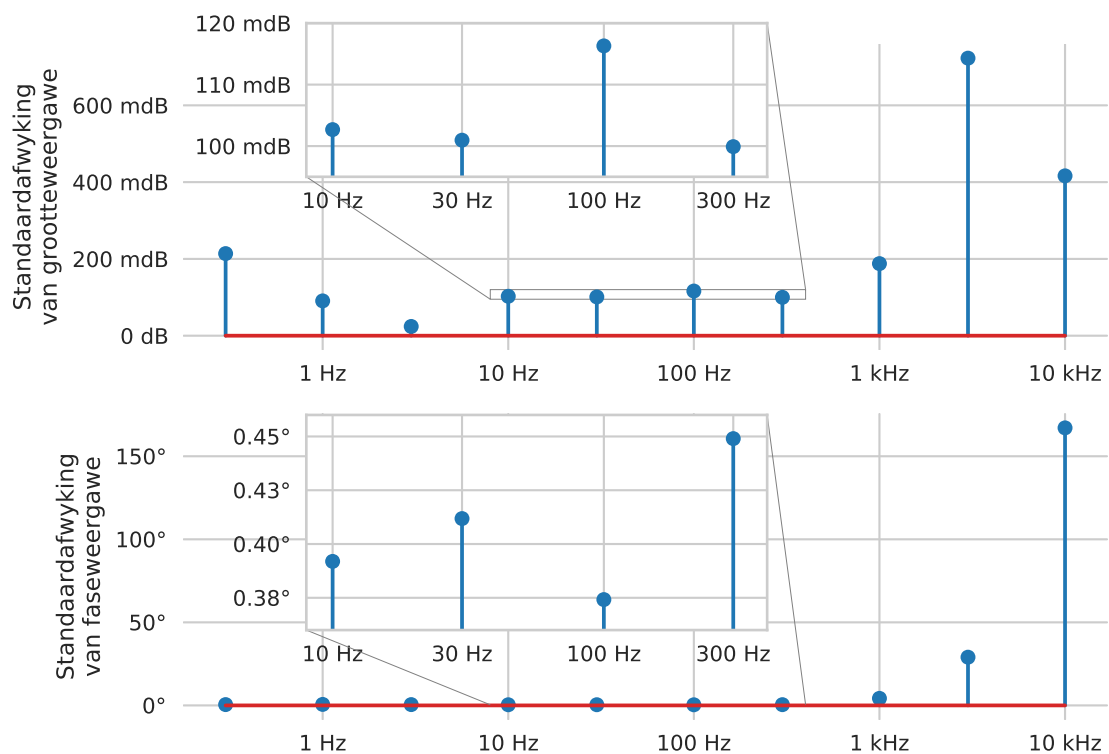
Figuur 5.10: Vervaardigde stroombaanbordjie vir die oudiokoppelvlak. Hierdie figuur is saamgestel deur albei kante van die bord te skandeer en dit met behulp van fotoredigeringsagteware (GIMP) bo-oor mekaar te plaas.



Figuur 5.11: Simulasie-opstelling van die voorversterker en filter in LTSpice.



Figuur 5.12: Gedetailleerde gesimuleerde en gemete frekwensieweergawes van een filterkanaal. 'n Vergelyking word getref tussen frekwensieweergawes met én sonder die voorversterker.



Figuur 5.13: Standaardafwyking van die gemete frekwensieweergawe van 5 verskillende kanale. Die weergawe van elke kanaal is gemeet tussen die *intree* van die voorversterker en die *uittree* van die filter.

Hoofstuk 6

Digitale oudiobuffer

Sodra 'n monster van die mikrofone versyfer is, moet die data van die versyferaar se monsterregister na bestendige geheue verskuif word, sonder om ooit 'n monster te verloor.

Aanvanklik is die versyferaar direk aan 'n **RPI 1A**¹ met 'n **Raspbian** bedryfstelsel gekoppel. 'n **Python**-program is geskryf om monsters via **SPI** van die versyferaar af te lees. Baie monsters het verlore gegaan, omdat die program gereeld deur die bedryfstelsel se taakskeduleerder onderbreek word. Daarna is dieselfde program in C geskryf en met hoogste prioriteit op die **RPI 1A** geloop. Sodoende het minder monsters verlore gegaan, maar dit was steeds nie aanvaarbaar nie – die **monstertempo** moet konstant wees, en geen monsters mag verlore gaan nie.

Die versyferaar en sagteware is toe na 'n **RPI 2B** geskuif, wat 'n vinniger, vier-kern verwerker het. Die vinniger verwerker het wel 'n verbetering gebied, maar monsters het nog steeds verlore gegaan. Die gebruik van 'n **RPI 3B** in dieselfde opstelling het geen verdere verbetering bo die 2B gebied nie.

'n Buffer word dus tussen die **RPI** en die versyferaar benodig. Die mees voor die hand liggende oplossing is om so 'n buffer met 'n mikrobeheerder te implementeer. Alternatiewelik sou 'n buffer met behulp van 'n **herprogrammeerbare logikahekskikking (FPGA, Field-Programmable Gate Array)** geïmplementeer kon word. Daar was egter nie genoeg tyd om die **FPGA**-oplossing te ondersoek nie.

6.1 SPI en DMA

Daar is verskeie maniere om data tussen mikrobeheerders oor te dra, waarvan **SPI** een van die eenvoudigste is. Daar is hoofsaaklik vier maniere om 'n mikrobeheerder se **SPI**-bus te gebruik:

Deur 'bit banging' As die mikrobeheerder nie 'n **SPI**-bus het nie, kan dit nagmaak word deur bisse direk na **GPIOs** te skryf. Hierdie opsie vereis die meeste verwerkertyd en is betreklik moeilik om te implementeer.

Blokkerend As daar wel 'n **SPI**-bus is, kan woorde² een op 'n slag na die **SPI**-bus se databuffer geskryf word. Na elke woord wat geskryf is, wag die sagteware in 'n lus totdat die **SPI**-bus 'n vlaggie stel wat aandui dat die transaksie voltooi is. Die verwerker bly dus besig (word geblokkeer) solank as wat **SPI**-transaksies uitgevoer word.

Onderbrekings-gedrewe Die meeste mikrobeheerders met **SPI**-busse het ook 'n onderbrekingsstelsel en kan gestel word om 'n **onderbrekingsversoek** te genereer sodra 'n **SPI**-transaksie voltooi is. Dit beteken dat die verwerker die greep na die databuffer kan skryf en met ander werk voort kan gaan totdat die transaksie voltooi is. Dan word die volgende greep geskryf, en die proses word herhaal.

¹ 'n Beta-weergawe wat in Desember 2011 vervaardig is.

² 'n Woord is gewoonlik 1–4 B, afhangend van die tipe mikrobeheerder.

DMA-gedrewe Regstreekse geheue-toegang (DMA, Direct Memory Access) is 'n manier om data outomaties uit een geheue-area na 'n ander te kopieer wanneer 'n vooraf-bepaalde sneller geaktiveer word. Hierdie sneller kan byvoorbeeld die onderbrekingsvlaggie van die **SPI**-bus wees. Dit gebeur sonder inbeslagneming van die verwerker. **DMA** kan dus gebruik word om outomaties die volgende greep uit 'n reeks vooraf-beplande **SPI**-transaksies na die **SPI**-databuffer te kopieer, sonder dat die verwerker op elke **onderbrekingsversoek** reageer. Wanneer al die transaksies voltooi is, genereer die **DMA**-bus self 'n **onderbrekingsversoek**, waarop die verwerker kan reageer om die resultate van al die afgelope **SPI**-transaksies op een slag te prosesseer.

Hierdie opsies speel 'n belangrike rol in die keuse van 'n mikrobeheerder vir hierdie projek.

6.2 PIC16F1847-mikrobeheerder

Microchip se PIC-mikrobeheerders is ondersoek³, omdat 'n **Pickit3** programmeerder⁴ beskikbaar was en die skrywer reeds ervaring met daardie familie van mikrobeheerders het.

Die eerste buffer-prototipe is gebaseer op die PIC16F1847, 'n 18-pen mikrobeheerder met twee selfstandige **SPI**-modules, 14 kB programgeheue, 1 kB datageheue en 'n interne 32 MHz ossillator. Die mikrobeheerder is met behulp van die Microchip MPLAB-X ontwikkelingsomgewing⁵, die **Pickit3** programmeerder en die XC8 C-kompileerder⁶ geprogrammeer.

Die PIC16F1847 benodig min eksterne komponente, wat die stroombaanontwerp baie eenvoudig maak.

In vergelyking met meer gevorderde mikrobeheerders soos die PIC32-reeks, reageer die PIC16F1847 baie vinnig op **onderbrekingsversoeke**. Volgens die datablad (Microchip, 2016, §8.2) word die hantering van onderbrekings met hoogstens 5 kloksiklusse vertraag. As die mikrobeheerder in C geprogrammeer word (eerder as **saamsteltaal**), is daar 'n ekstra vertraging wanneer die programkonteks gestoor word⁷. Hoe minder registers 'n mikrobeheerder het, hoe vinniger is die oorskakeling van een programkonteks na 'n ander.

Die maksimum klokspoed van die **SPI**-bus in meester-modus is $F_{osc}/4 = 32 \text{ MHz}/4 = 8 \text{ MHz}$ (Microchip, 2016, §25.2.3), wat net minder is as die maksimum klokspoed van 10 MHz vir die **SPI**-bus van die versyferaar (Microchip, 2011b, p. 1).

Hierdie mikrobeheerder het egter nie 'n **DMA**-bus nie, en die **SPI**-buffer is slegs 1 B lank. Dit beteken dat die verwerker onderbreek moet word om data te lees en te skryf vir elke greep wat oor **SPI** verstuur word.

Die **SPI**-transaksie met die versyferaar moet binne die monsterperiode voltooi word, sodat die register nie deur die volgende monster oorskryf word nie. Die PIC is die **SPI**-meester (wat die transaksie moet begin) en moet dus gereed wees om op die **onderbrekingsversoek** van die versyferaar te reageer. Die **RPI** kan slegs as 'n **SPI**-meester optree, wat beteken dat die PIC gereed moet wees om transaksies deur die tweede **SPI**-bus te hanteer wanneer die **RPI** dit versoek. As die PIC laat is vir die transaksies met die versyferaar, gaan monsters verlore, en as die PIC laat is vir transaksies met die **RPI**, word korrupte data verstuur.

Met 6 mikrofoonkanale en 24-bis resolusie is elke monster 18 B groot. Dit is ook wenslik om vir elke monster 'n akkurate tydstempel te gee, wat minstens 4 B per monster in beslag sal neem. Gevolglik kan hierdie mikrobeheerder in die beste geval minder as $1024/22 \approx 46$ monsters, oftewel 9,5 ms se data stoor. Dit sou beteken dat die **RPI** elke 9,5 ms genoeg data van die PIC oor **SPI** moet ontvang om die buffer leeg te maak, anders gaan monsters verlore.

³ By <http://www.microchip.com/maps/> kan Microchip-komponente volgens spesifikasies uitgesoek word.

⁴ Beschikbaar by www.microchip.com/pickit3.

⁵ Beschikbaar by <http://www.microchip.com/mplab/mplab-x-ide>.

⁶ Beschikbaar by www.microchip.com/mplab/compiler.

⁷ Al die registers word op die programstapel geberg, sodat daar later teruggekeer kan word na die funksie wat besig was om uitgevoer te word toe die onderbreking plaasgevind het.

#	Naam	Funksie
10, 26 en 38	VDD	Positiewe toevoerspanning
9, 25 en 41	VSS	GND
19	AVDD	Positiewe verwysingspanning vir die ingeboude versyferaar
20	AVSS	Negatiewe verwysingspanning vir die ingeboude versyferaar

Tabel 6.1: Verbindings van toevoerspanning en **GND** aan die mikrobeheerder.

Daar is egter 'n verdere beperking, naamlik dat die **SPI**-transaksies tussen die **RPI** en die PIC korter as een monsterperiode ($208,33\mu\text{s}$) moet wees, sodat die mikrobeheerder nie 'n transaksie met die versyferaar misloop nie. Boonop neem die veranderlikes wat deur die program gebruik word 'n deel van die datageheue in beslag, sodat die volle 1 kB nie gebruik kan word om data te buffer nie. 'n Vinniger mikrobeheerder met **DMA** en meer geheue word dus benodig.

6.3 PIC32MX-mikrobeheerder

Diligent vervaardig 'n reeks mikrobeheerder-ontwikkelingsborde genaamd ChipKIT. Die ChipKIT uC32⁸ is op die Arduino-ontwikkelingsbord gebaseer en bevat 'n PIC32MX340F512H-mikrobeheerder⁹. Hierdie mikrobeheerder het 'n klokspoed van 80 MHz, 512 kB programgeheue, 32 kB datageheue, 4 **DMA**-kanale en 2 **SPI**-busse. Twee **DMA**-kanale word vir elke **SPI**-bus benodig (een om uit die **SPI**-buffer te lees en een om na die **SPI**-buffer te skryf). As elke 18 B monster saam met 'n 6 B tydstempel gestoor word, sal die 32 kB datageheue hoogstens $\frac{32\text{B} \times 1024}{18\text{B} + 6\text{B}} \approx 1365$ monsters kan stoor, wat teen 'n monstertempo van 4800 Hz ongeveer 284 ms se data verteenwoordig. Die werklike bufferlengte word in §6.7 bespreek en bereken.

Die ChipKIT uC32 ontwikkelingsbordjie is gebruik om die konsep te toets. Daarna is 'n eie mikrobeheerderbordjie wat net die nodige verbindings het en heelwat goedkoper is, ontwerp. Die bufferstroombaai is met KiCad uitgelê en na PCBWay gestuur vir vervaardiging. Figuur 6.1(a) toon die resultaat. Alle komponente is met die hand gesoldeer. Die bordjie is ontwerp om bo-op 'n **RPI** ingeprop te word, soos aangedui in Figuur 6.1(b).

Die PIC32MX word gebruik om data betyds via **SPI**-bus 1 van die versyferaar af te lees, dit tydelik in geheue te berg en dit in die regte volgorde via **SPI**-bus 2 aan die **RPI** te stuur. Die werking van hierdie bordjie en die kommunikasie met die **RPI** word in die res van hierdie hoofstuk noukeuriger beskryf.

6.4 Verbindings

Die volledige stroombaai vir die mikrobeheerder is te groot om hier te toon. In die plek daarvan word die komponentdiagram in Figuur 6.2 gegee, en 'n beskrywing van elke verbinding volg hieronder. 'n Volledige stroombaandiagram is op DVD E.1 in Bylae E beskikbaar.

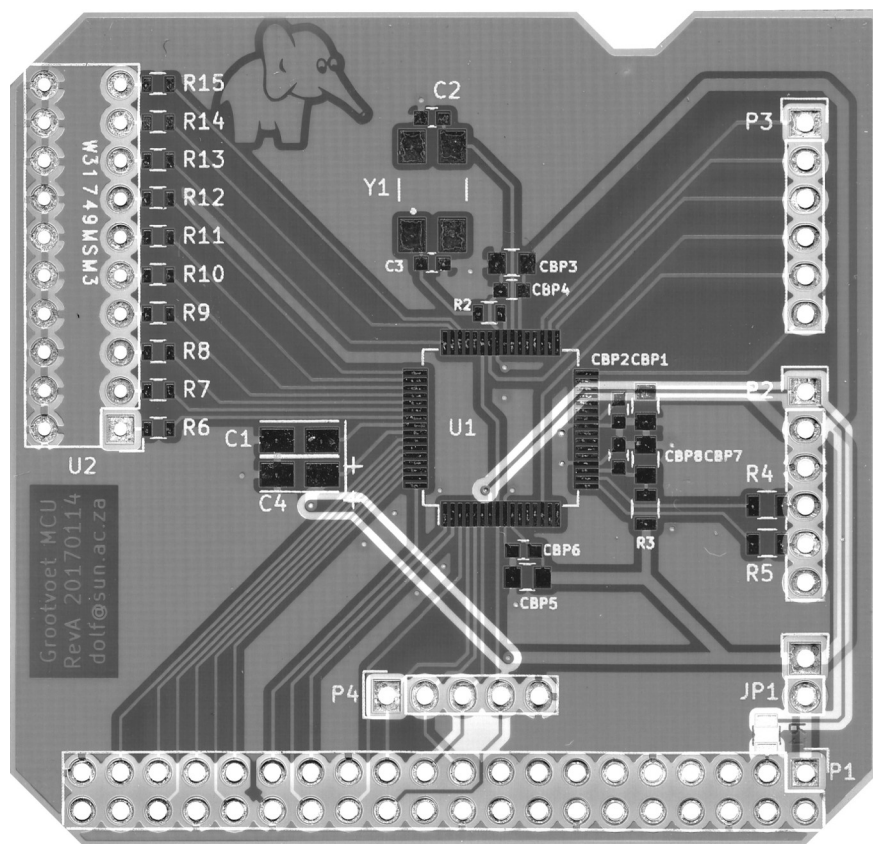
Alle pennommers in hierdie afdeling is soos op die PIC32MX340F512H-skyfie se datablad (Microchip, 2011c, bl. 7). Die stroombaauitleg en name van komponente en verbindingsrye word in Figuur 6.1(a) getoon.

Die mikrobeheerder het verskeie penne waar die toevoerspanning gekoppel moet word. Hierdie verbindings word in Tabel 6.1 getoon.

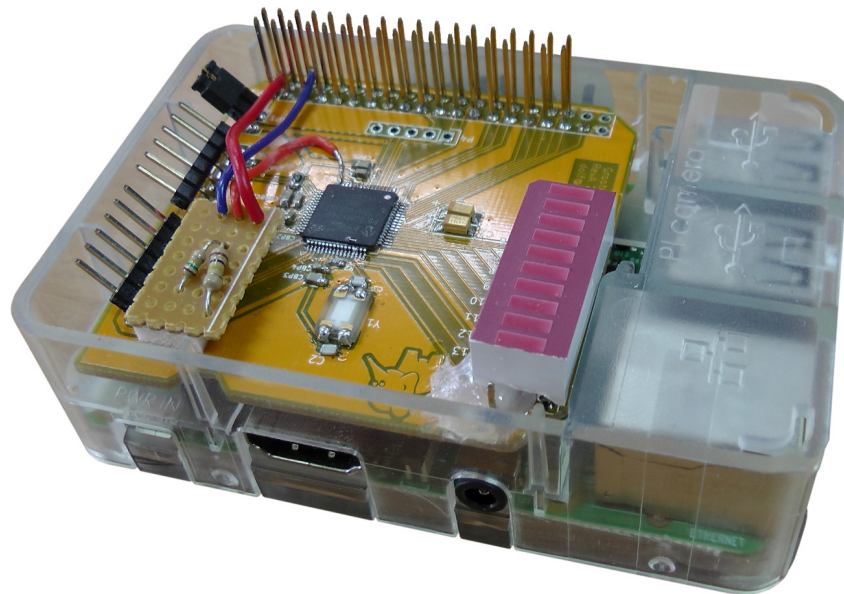
Volgens die aanbevelings in die datablad (Microchip, 2011c, §2.2) word twee keramiekkapasitors (100 nF en 1–10 nF) in parallel tussen elke stel VDD- en VSS-penne geplaas. Albei kapasitors

⁸ Sien www.microchip.com/chipKITu32DevelopmentBoard5782.

⁹ Sien <http://www.microchip.com/design-centers/32-bit/architecture/pic32mx-family>.

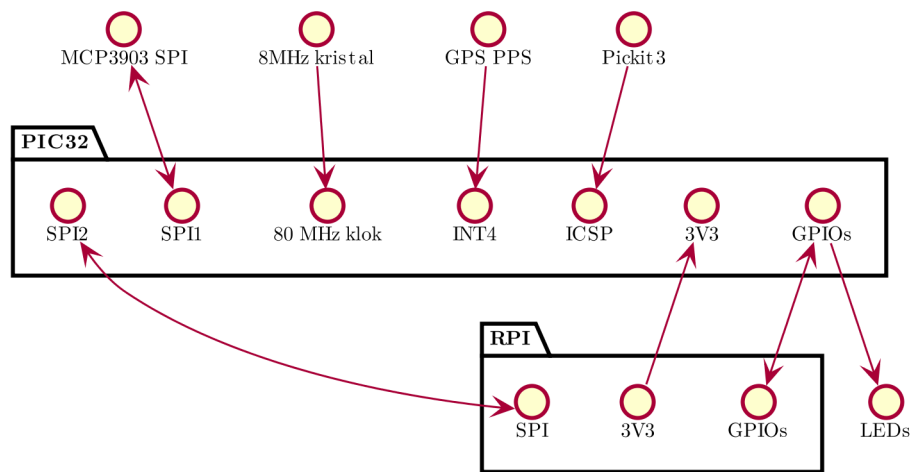


(a) Die stroombaanbordjie sonder komponente. Hierdie figuur is saamgestel deur albei kante van die bord te skandeer en dit met behulp van fotoredigeringsagteware (GIMP) bo-oor mekaar te plaas.

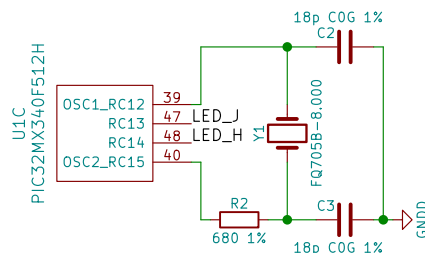


(b) Die stroombaanbordjie met komponente, bo-op 'n **RPI** geïnstalleer.

Figuur 6.1: Vervaardigde buffer-stroombaanbordjie. Die verbindings met elke pen van die mikrobeheerder word in Tabel 6.2 gegee en in §6.4 bespreek. Die verbindings van ry P1 met die **RPI** word ook in §7.1 behandel.



Figuur 6.2: 'n Opsomming van alle verbindings van die PIC32MX. **SPI1** word gebruik om monsters een vir een van die versyferaar te lees, sodra hulle gereed is. **SPI2** word gebruik om monsters op aanvraag in groepe na die **RPI** te stuur. 'n 8 MHz kristal beheer die verwerker se klok. Onderbrekingspen **INT4** word gebruik om **PPS**-seine van die **GPS** module te ontvang sodat akkurate tydtempels op elke monster geplaas kan word. Die **ICSP**-poort word gebruik om die mikrobeheerder te programmeer. Die 3,3 V-lyn van die **RPI** word gebruik om die mikrobeheerder aan te dryf. Tabelle 6.1 en 6.2 bied 'n noukeuriger beskrywing van elke verbinding. Al die komponente verwys na dieselfde **GND**.



Figuur 6.3: Ossilator-stroombaan vir die mikrobeheerder.

moet 'n lae serie-weerstand en 'n resonansie bó 20 MHz hê. Vir alle praktiese doeleindes voldoen die meeste X7R-kapasitors aan hierdie vereistes (CDE Cornell Dubilier, 2017).

Pen 57 (**ENVREG**) word aan **VDD** gekoppel sodat die mikrobeheerder sy interne spannings-reguleerder gebruik. 'n 10 μ F, 10 V tantaal-kapasitor word tussen pen 56 (**VCORE**) en **GND** geplaas om die reguleerder se spanning stabiel te hou (Microchip, 2011c, §2.3).

Pen 7 is 'n laag-aktiewe herstelpen, genaamd **nMCLR**. Die pen word deur 'n 100 Ω -weerstand (**R1**) aan **GPIO2** van die **RPI** gekoppel. Die pen is ook direk blootgestel op ry P2, wat met die **Pickit3** programmeerder verbind word tydens ontwikkeling. Die weerstand keer dat die **RPI** die mikrobeheerder herstel terwyl dit geprogrammeer word. Dit keer ook konflik tussen die **RPI** en **Pickit3** – daar sal hoogstens $\frac{V_{DD}-V_{SS}}{100\Omega} = 33\text{ mA}$ vloei wanneer een kant die **nMCLR**-lyn hoog probeer maak – en die ander, laag. 'n Optrekweerstand op die **nMCLR**-pen is nie nodig nie, want dit word aan 'n **uittreep** van die **RPI** gekoppel, wat dit óf hoog, óf laag sal dryf.

Penne 17 en 18 (**ICSPCLK** en **ICSPDAT**) kan direk aan die programmeerder gekoppel word, maar 10 Ω serie-weerstande (**R4** en **R5**) word gebruik om hulle teen elektrostatische ontlading te beskerm (Microchip, 2011c, §2.5). Hierdie penne is ook op ry P2 beskikbaar.

'n Eksterne 8 MHz-kristal (**Y1**) word as 'n ossilator vir die mikrobeheerder gebruik (sien ook Figuur 6.3). Die datablad bied noukeurige spesifikasies vir die uitleg van die ossilator-stroombaan (Microchip, 2011c, §2.8). Die uiteindelijke uitleg is middel bo in Figuur 6.1(a) te sien.

Tabel 6.2 toon die verbindings tussen die mikrobeheerder en die res van die projek, soos wat dit voorkom op die prototipes wat vir hierdie projek gebou is. Dit stem ooreen met die borduitleg

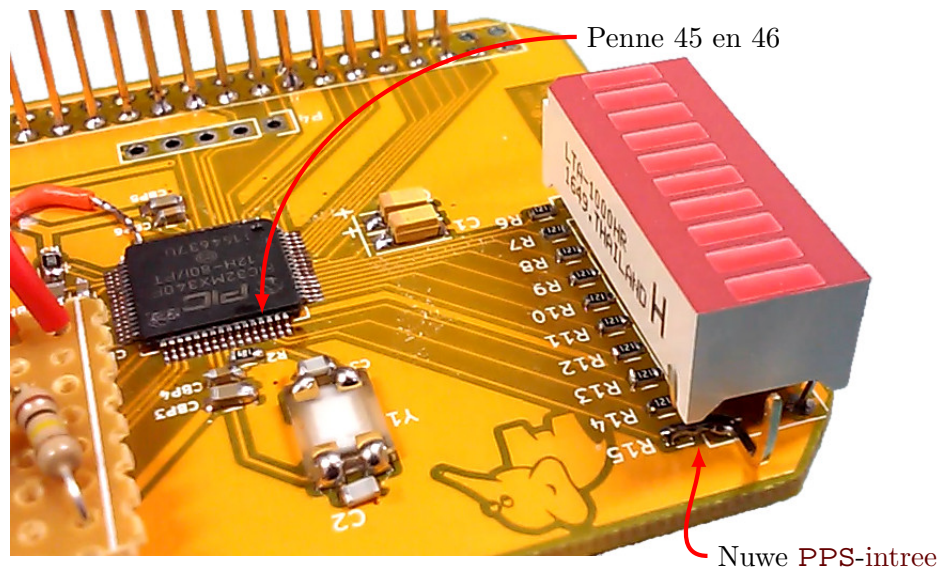
#	Poort	In/Uit	Funksie	Verbind met	Figuur
1	RE5	U	–	RPI GPIO1	–
2	RE6	U	–	RPI GPIO0	–
3	RE7	I	–	RPI GPIO7	–
4	RG6	I	SCK2	RPI GPIO11 SCLK	–
5	RG7	I	SDI2	RPI GPIO10 MOSI	–
6	RG8	U	SDO2	RPI GPIO9 MISO	–
7	nMCLR	I	PIC32 nRESET	RPI GPIO2	–
8	RG9	I	SS2	RPI GPIO8 CE0#	–
14	RB2	U	SS1	MCP3903 nSELECT	5.1
32	RF5	U	MCP3903 nRESET	MCP3903 nRESET	5.1
33	RF3	U	SDO1	MCP3903 MOSI	5.1
34	RF2	I	SDI1	MCP3903 MISO	5.1
35	RF6	U	SCK1	MCP3903 SCLK	5.1
39	RC12	–	OSC1	–	6.3
40	RC15	–	OSC2	–	6.3
42	RD8	I	INT1	MCP3903 nREADY	5.1
43	RD9	I	INT2	RPI GPIO8 CE0#	–
46	RD0	U	–	LED K	–
47	RC13	U	–	LED J	6.3
48	RC14	U	–	LED H	6.3
49	RD1	U	–	LED G	–
50	RD2	U	–	LED F	–
51	RD3	U	–	LED E	–
52	RD4	U	–	LED D	–
53	RD5	U	–	LED C	–
54	RD6	U	–	LED B	–
55	RD7	U	–	LED A	–
58	RF0	U	MCU_ERROR	RPI GPIO26	–
59	RF1	U	SPI_READY	RPI GPIO19	–
60	RE0	U	–	RPI GPIO16	–
61	RE1	U	–	RPI GPIO13	–
62	RE2	I	–	RPI GPIO12	–
63	RE3	I	–	RPI GPIO6	–
64	RE4	I	–	RPI GPIO5	–

Tabel 6.2: Verbindings van die buffer-mikrobeheerder, soos vervaardig. Penne wat nie aan enigiets gekoppel is nie, word weggelaat. Penne sonder 'n funksie is slegs gekoppel vir ontwikkelingsdoeleindes.

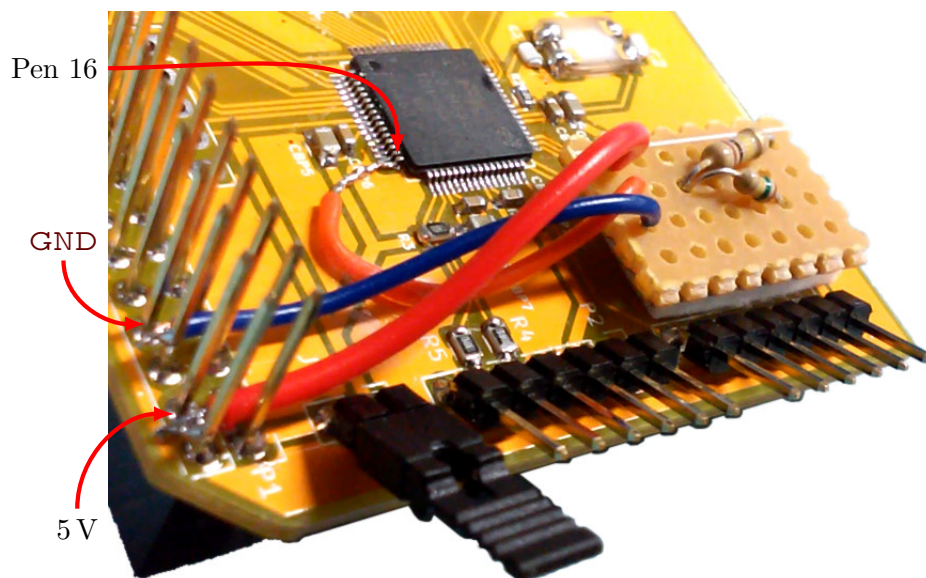
wat in Figuur 6.1(a) getoon word.

Twee funksies – 'n **GPS**-module (sien §8) en 'n spanningsmonitor (sien §6.10) – is egter later bygevoeg, wat 'n herontwerp van die bordjie vereis. Daar was egter nie tyd om die nuwe ontwerp te laat vervaardig nie, en 'n paar aanpassings is met die hand aan elke bordjie gemaak:

- **Liguitstralende diode (LED, Light-Emitting Diode)** K is van pen 46 ontkoppel, en penne 45 en 46 van die PIC32 is aan mekaar verbind. Albei is as **intree**poorte gestel, met pen 45 as 'n onderbrekings-pen (INT4). Die stroombaanspoor wat na die **LED** geloop het, is gebruik om die **GPS PPS**-sein te koppel (sien Figuur 6.4).
- Pen 16 word as 'n analoog**intree** gebruik om die toevoerspanning na die **RPI** te monitor. Om dit te kan doen, word 'n spanningsdeler tussen die 5 V-pen en **GND** geplaas (sien Figuur 6.5). Die spanningsmonitor word in meer detail in §6.10 beskryf.



Figuur 6.4: Om pulse vanaf die **GPS PPS**-pen te ontvang, moet 'n onderbrekingspen gebruik word. Daar is geen spoor aan pen 45 (INT4) verbind nie. Penne 45 en 46 is aan mekaar vasgesoldeer, en die spoor van pen 46 is gebruik. Die **LED** wat aan pen 46 gekoppel was, is ontkoppel.



Figuur 6.5: Om die batteryspanning te meet, word 'n spanningsdeler tussen die 5 V-pen en **GND** geplaas. Die gedeelde spanning word dan deur die analogoointreepoort op pen 16 van die PIC32MX versyfer.

#	Poort	In/Uit	Funksie	Verbind met	Figuur
16	RB0	I	PIC32 Versyferaar	5 V-toevoerspanning	–
30	RB15	U	MCP3903 nRESET	MCP3903 nRESET	5.1
31	RF4	I	U2RX	GPS TX	–
32	RF5	U	U2TX	GPS RX	–
45	RD11	I	INT4	GPS PPS	–
61	RE1	U	BATTERY_LAAG	RPI GPIO13	–

Tabel 6.3: Verbeterde verbindings van die buffer-mikrobeheerder. Penne wat dieselfde bly as in Tabel 6.2 of nie meer aan enigiets gekoppel is nie, word uitgelaat.

Verdere verbeterings kan in die toekoms aan hierdie ontwerp aangebring word:

- Die reeks **LEDs** wat aan penne 46–55 gekoppel is, was nuttig vir ontfouting, maar kan in 'n volgende weergawe verwyder word.
- Pen 60 is gebruik as **SPI ACK**, maar het in onbruik verval met nuwer weergawes van die kommunikasieprotokol (§6.6) tussen die **RPI** en die PIC32 en kan dus ontkoppel word.
- MCP3903 nReset kan van pen 32 na pen 30 verskuif word sodat die **universele asinkrone ontvanger en sender (UART, Universal Asynchronous Receiver-Transmitter)** op penne 31–32 gebruik kan word om met die **GPS**-module te kommunikeer. In die huidige ontwerp kommunikeer die **GPS UART** slegs met die **RPI** en word die tyd van die **GPS** deur die **RPI** na die PIC32 oorgedra. Dit word in meer detail in §8 bespreek.
- Penne 1–3 en 61–64 is ook aan die **RPI** gekoppel vir ingeval dit nodig sou wees om ekstra kommunikasie tussen die **RPI** en PIC32 te hê. Pen 61 is gebruik as 'n **uittreepoort** wat aandui of die toevoerspanning hoog of laag is, maar die ander word nie gebruik nie. Die ont koppeling van die ander penne kan die borduitleg aansienlik vereenvoudig.

Tabel 6.3 toon die verbindings vir die voorgestelde verbeterings. Hierdie nuwe ontwerp is egter nog nie vervaardig of getoets nie.

Mikrobeheerderpennet wat nie gekoppel is nie, word as **uittreepenne** gestel en laag gedryf volgens die aanbeveling in [Microchip \(2011c, §2.10\)](#).

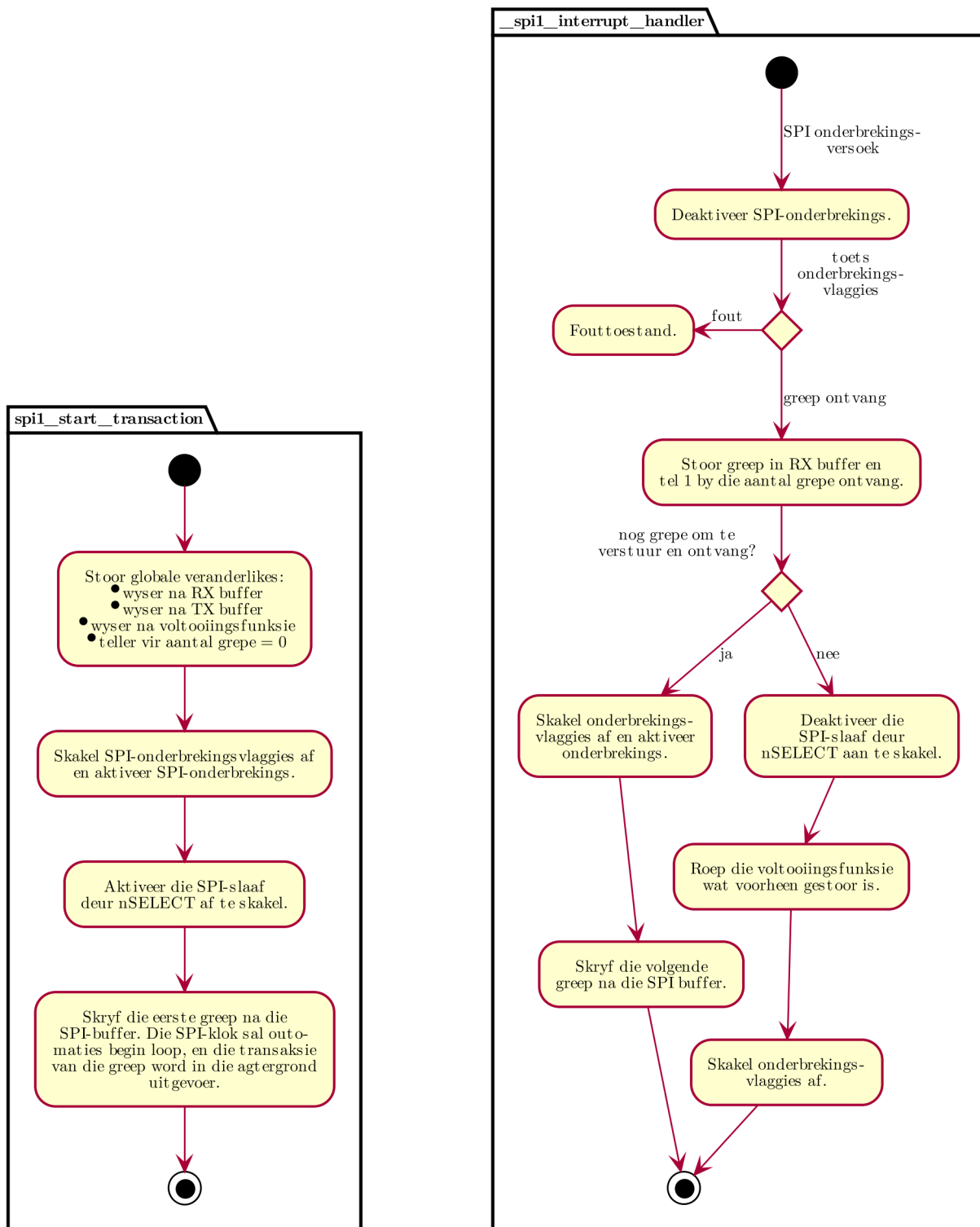
6.5 SPI-kommunikasie met versyferaar

SPI-bus 1 word gebruik om met die versyferaar te kommunikeer. 'n Pasgemaakte drywer is hiervoor geskryf. Die drywer bestaan uit drie funksies en twee **onderbrekingsroetines**. Die bronkode is op DVD E.1 in Bylae E beskikbaar.

Funksie `spi1_setup` berei die **SPI**-bus en **DMA**-busse voor vir gebruik. Dit moet uitgevoer word voordat enige van die ander `spi1_`-funksies geroep word. Die ander funksies word deur Figure 6.6 en 6.7 beskryf.

Die drywer kan dus op twee maniere gebruik word: a) Sonder **DMA**, maar met 'n onderbreking na elke greep. b) Met **DMA**, sodat daar slegs 'n onderbreking aan die einde van die totale transaksie plaasvind. Daar is nie 'n implementasie vir blokkerende **SPI**-transaksies nie.

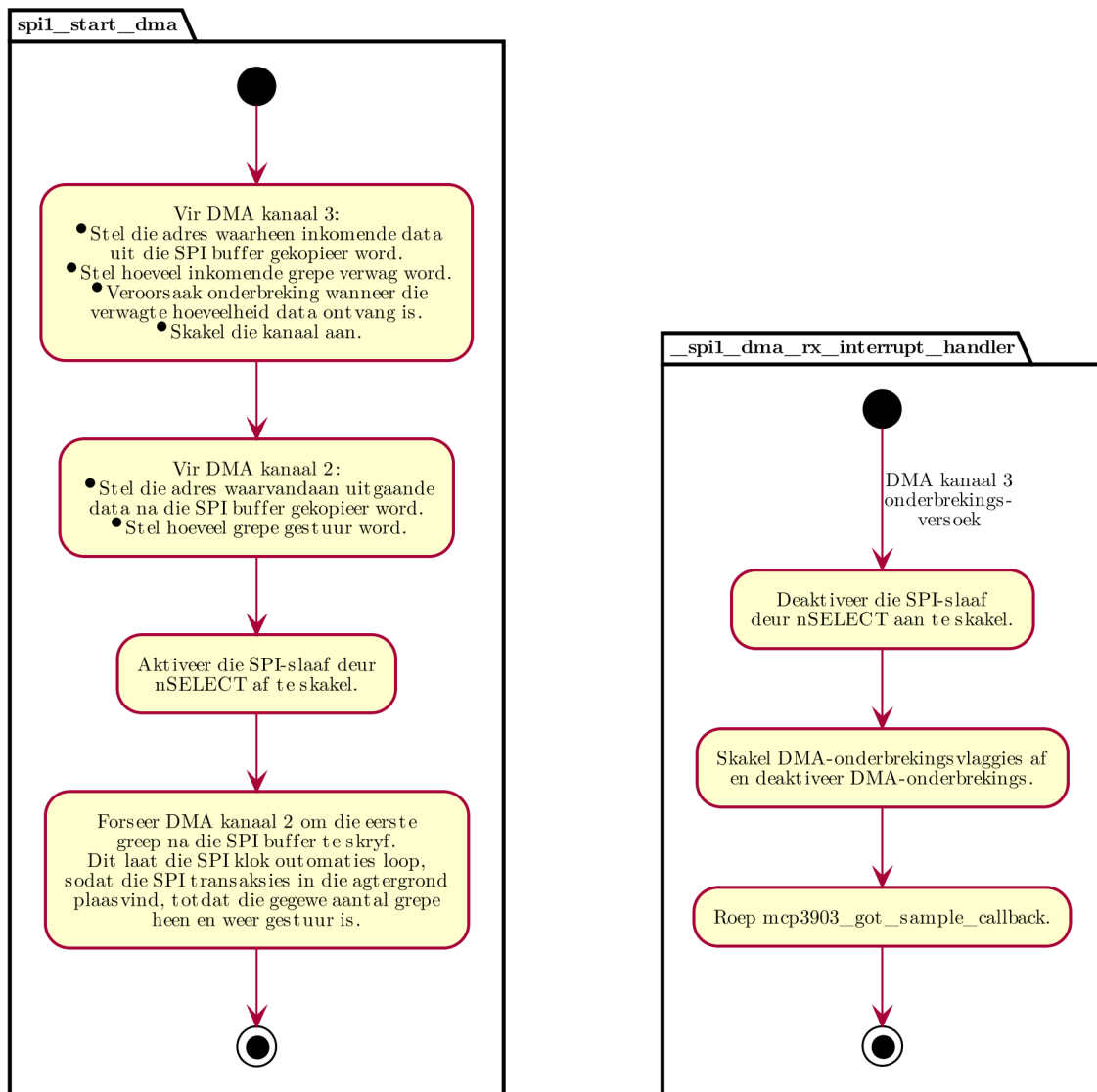
Figuur 6.8 toon 'n sekvensdiagram vir die interaksie tussen die versyferaar en die mikrobeheerder.



(a) Funksie `spi1_start_transaction` aktiveer die versyferaar en begin 'n SPI-transaksie wat een greep op 'n slag tussen die PIC32MX en MCP3903 uitruil. Na elke greep sal `_spi1_interrupt_handler` uitgevoer word.

(b) Onderbrekingsroetine `_spi1_interrupt_handler` word uitgevoer wanneer 'n fout plaasvind, of wanneer 'n greep tussen die PIC32MX en MCP3903 uitgeruil is, nadat 'n transaksie met `spi1_start_transaction` begin is.

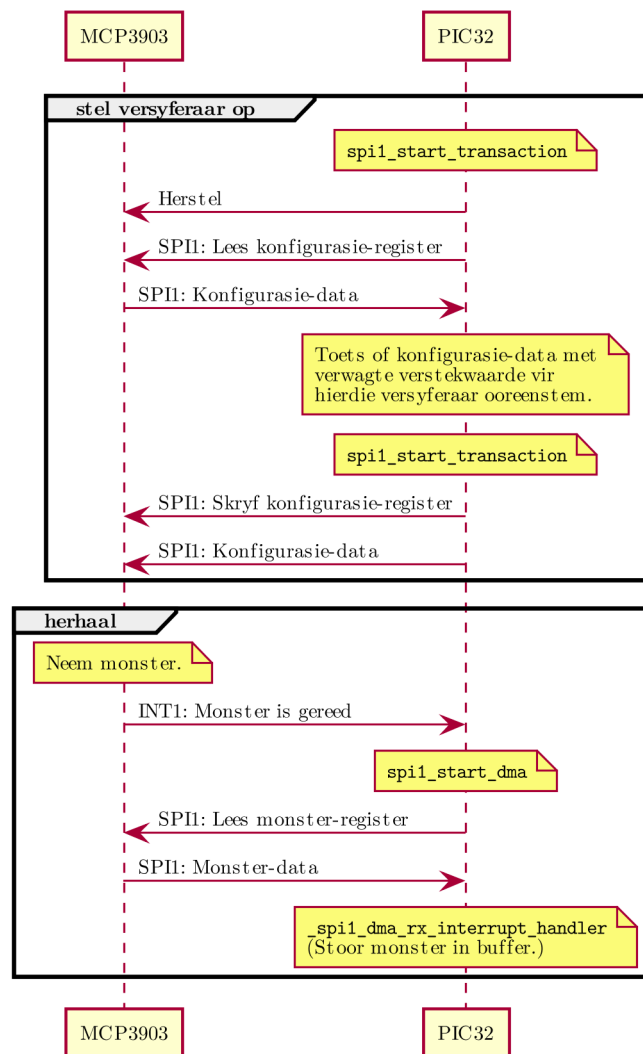
Figuur 6.6: Funksies vir onderbrekings-gedrewe SPI-kommunikasie tussen die mikrobeheerder en versyferaar. Hierdie twee funksies word op die PIC32MX uitgevoer, wat as meester van die SPI-bus optree. Dit word slegs gebruik in die aanvanklike opstelling van die versyferaar (sien Figuur 6.8).



(a) Funksie `spi1_start_dma` aktiveer die versyferaar en stel twee **DMA**-kanale op om 'n reeks **SPI**-transaksies in die agtergrond uit te voer. Nadat al die transaksies voltooi is, word `_spi1_dma_rx_interrupt_handler` een keer uitgevoer.

(b) Die **onderbrekingsroetine** `_spi1_dma_rx_interrupt_handler` word uitgevoer sodra 'n reeks transaksies wat met `spi1_start_dma` begin is, voltooiing bereik het.

Figuur 6.7: Funksies vir **DMA**-gedrewe **SPI**-kommunikasie tussen die mikrobeheerder en versyferaar. Hierdie twee funksies word op die PIC32MX uitgevoer, wat as meester van die **SPI1**-bus optree. Figuur 6.8 toon waar dit in die kommunikasieprotokol inpas.



Figuur 6.8: Sekwensdiagram vir die interaksie tussen die MCP3903 versyferaar en PIC32MX mikrobeheerder. Dit lees van bo na onder.

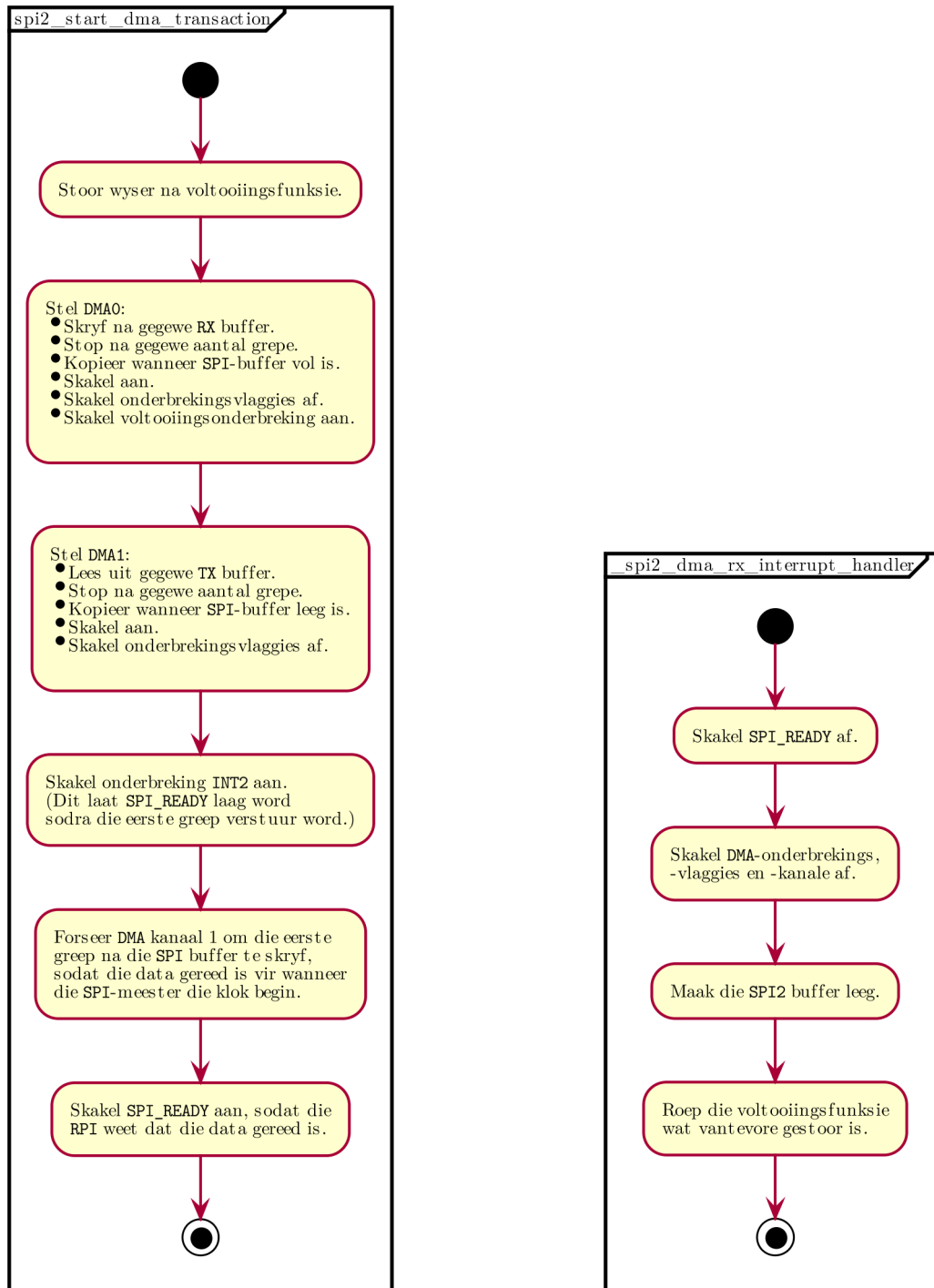
6.6 SPI-kommunikasie met RPI

SPI-bus 2 word gebruik om met die **RPI** te kommunikeer. Net soos in §6.5, is 'n pasgemaakte drywer hiervoor geskryf. Die drywer bestaan uit twee funksies en 'n **onderbrekingsroetine**. Die bronkode is op DVD E.1 in Bylae E beskikbaar.

Funksie `spi2_setup` berei die **SPI**-bus en **DMA**-busse voor vir gebruik en moet uitgevoer word voordat enige van die ander `spi2_`-funksies geroep word. Figuur 6.9 beskryf die res van die funksies.

Anders as in die vorige hoofstuk, waar die **SPI**-protokol deur die datablad van die MCP3903 bepaal word, moet daar 'n nuwe protokol geskep word vir kommunikasie tussen die **RPI** en die buffermikrobeheerder. Tabel 6.4 som hierdie protokol op. Alhoewel **SPI** 'n veldupleks kommunikasiemiddel is, is dit baie eenvoudiger om dit as halfdupleks te gebruik. Eers word 'n opdrag met parameters verstuur (5 grepe). Afhangend van die opdrag, word 'n antwoord met 'n spesifieke lengte terugverwag.

'n Ekstra **GPIO**-pen genaamd `SPI_READY` word gebruik om aan te dui wanneer die mikrobeheerder die antwoord klaar voorberei het. Sodra 'n opdrag ontvang word, daal die `SPI_READY`-pen, en dan wag die **RPI**, wat as **SPI**-meester optree, vir die pen om te styg voordat die **SPI**-klok weer begin, en die antwoord verstuur word.



(a) Funksie `spi2_start_dma_transaction` berei twee DMA-kanale voor om op die SPI-meester te kan reageer sodra die meester die SPI-klok laat begin. Die SPI_READY-lyn word gebruik om aan te dui wanneer die mikro-beheerder gereed is vir die transaksie.

(b) Die onderbrekingsroetine `_spi2_dma_rx_interrupt_handler` word uitgevoer sodra 'n reeks transaksies wat met `spi2_start_dma_transaction` begin is, voltooiing bereik het.

Figuur 6.9: Funksies vir DMA-gedrewe SPI-kommunikasie tussen die RPI en die PIC32MX. Hierdie twee funksies word op die PIC32MX uitgevoer, wat as slaaf van die SPI2-bus optree.

Opdrag	Parameter (4 B)	Beskrywing	Antwoord
0xF0	Aantal monsters	Haal monsters uit die buffer.	24 B per monster
0xF1	–	Kry die buffertoestand en herstel tellers.	0xGGHHHHIIJJJJ, waar: G=0xA5 (ID) H=Buffer-lengte I=Aantal verlore monsters a.g.v. stadige SPI1. J=Aantal verlore monsters a.g.v. oorvol buffer.
0xE0	–	Herstel die buffer en alle tellers.	0x0F (ACK)
0xE1	Sekondes sedert die Unix-epog.	Stel die tyd.	0x0F (ACK)
0xE2	–	Kry die tyd.	0xGGGGGGGGHHHH, waar: G=Sekondes sedert die Unix-epog. H=Hoeveel monsters hierdie sekonde ontvang is.

Tabel 6.4: Protokol vir SPI-kommunikasie tussen die buffer-mikrobeheerder en die RPI. Vir elke opdrag wat via SPI2 van die RPI ontvang kan word, is daar 'n spesifieke formaat waarin die antwoord teruggestuur moet word.

Opdrag 0xF1 word gebruik om te toets of die buffermikrobeheerder wel aan die RPI gekoppel is. Die ID (0xA5) is 'n arbitrêre, vooraf vasgestelde getal.

Opdragte wat nie 'n antwoord terug verwag nie, kry in elk geval 'n ACK (0x0F) om aan te dui dat die opdrag ontvang en uitgevoer is. Dit is nodig sodat 'n sukses onderskei kan word van die geval waar die mikrobeheerder 'n fouttoestand betree.

Omdat daar geen inherente manier is om te weet of die data wat oor 'n SPI-bus ontvang word korrek is nie, word 'n foutkorreksiekode gebruik. Daar is hoofsaaklik twee tipes foutkorreksiekodes, naamlik *a)* terugwaartse en *b)* voorwaartse foutkorreksiekodes. Terugwaartse kodes laat die ontvanger slegs toe om te weet of die ontvangde data korrek is. Voorwaartse kodes stel die ontvanger in staat om die ontvangde data te korrigeer, mits dit minder as 'n sekere aantal foute het. Vir hierdie projek word 'n Reed-Solomon (RS) kode gebruik (Reed en Solomon, 1960), wat die volgende voordele bied:

- Dit is effektief in gevalle waar foute in sarsies voorkom (Reed en Solomon, 1960, p. 301). Dit wil sê: Die kans dat 'n boodskap met baie opeenvolgende foute as foutief herken sal word, is hoog.
- Die kode word aan die einde van die oorspronklike data bygevoeg, terwyl die oorspronklike data onveranderd bly. As die ontvanger eers vasgestel het dat die data korrek is, hoef daar dus geen verdere dekodering plaas te vind nie.
- As M grepe by die verstuurde data gevoeg word, kan M foute by bekende posisies, of $M/2$ foute by onbekende posisies reggestel word (Minsky, 2009).

Die RS-implementasie vir hierdie projek is op oopbron-kode deur Minsky (2013) gebaseer. Die skrywer het dit vir hierdie projek aangepas om met die XC8 C-kompileerder op die PIC32 te werk. In beide rigtings word twee RS-grepe by elke SPI-transaksie gevoeg.

6.7 Draadbestande sirkelbuffers

'n Buffer is 'n waglys met eindige lengte, oftewel 'n skikking wat volgens die *eerste-in, eerste-uit* (**FIFO, First In, First Out**) beginsel werk. Op 'n mikrobeheerder moet 'n aaneenlopende geheue-blok vooraf hiervoor toegeken word, omdat dit vinniger en eenvoudiger is as om geheue dinamies (soos met 'n *aaneengeskakelde lys*) toe te ken. As 'n buffer op so 'n wyse geïmplementeer word dat die lees- en skryfindeks (onderskeidelik vir *stoot-* en *haal*bewerkings), weer na die begin van die geheue-blok spring nadat dit die einde bereik het, word dit 'n sirkelbuffer genoem.

As meer as een *draad* (byvoorbeeld die hooflus en een of meer *onderbrekingsroetines*) van die buffer gebruik wil maak, moet die buffer op so 'n manier geïmplementeer word dat daar nie 'n onderbreking kan plaasvind terwyl 'n element bygevoeg of verwyder word nie.

- *Wedersydse uitsluitings* kan gebruik word om seker te maak dat slegs een *draad* op 'n slag toegang tot die buffer verkry. *Wedersydse uitsluitings* mors egter tyd, omdat *drade* vir mekaar moet wag. Dit kan ook veroorsaak dat die stelsel vries as twee *drade* vir mekaar sou bly wag. Dit verhoog dus ook die kompleksiteit van die sagteware.
- Die probleem kan ook opgelos word deur te verseker dat die bewerkings op die buffer atomies is. Of dit moontlik is, hang van die instruksiestel van die verwerker af.
- Kwasi-atomiese bewerkings kan geskep word deur alle onderbrekings te deaktiveer, meer as een instruksie uit te voer en daarna weer die onderbrekings toe te laat. Dit vertraag egter die hantering van *onderbrekingsveroeke*.

Alternatiewelik moet 'n slimmer implementasie gebruik word, waar dit nie saak maak as 'n onderbreking plaasvind terwyl 'n element *gestoot* of *gehaal* word nie.

As daar slegs een *draad* is wat ooit elemente probeer *stoot* (die skrywer) en een wat elemente *haal* (die leser), kan die implementasie baie vereenvoudig word. In hierdie projek is die *onderbrekingsroetine* vir **DMA**-kanaal 3 (Sien §6.5) die skrywer, en die hooflus is die leser. Die bewerkings vir byvoeging en verwydering van elemente word in Figuur 6.10 getoon.

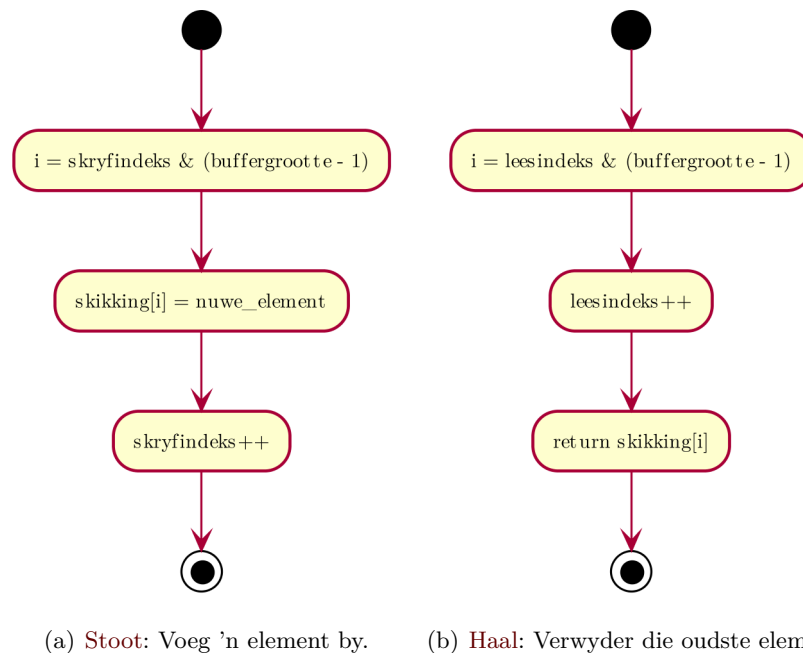
Toetse om te kyk of die buffer leeg of vol is, word aan die gebruiker oorgelaat, sodat daar geen vertakkings (oftewel besluitneming) in hierdie bewerkings plaasvind nie. Verder word die skryfindeks en leesindeks apart gestoor, sodat manipulasie van die een nie die ander beïnvloed nie. Omdat daar geen vertakkings is nie en daar slegs een skrywer is, sal 'n onderbreking waarin 'n *haal*-bewerking plaasvind, nie 'n toestand veroorsaak waarin die *stoot*-bewerking sal faal nie. In 'n buffer-implementasie waar die leesindeks en lengte gestoor word en die skryfindeks daarvan afgelei word, sal die *stoot*-bewerking na 'n verkeerde indeks skryf indien 'n *onderbrekingsroetine* tydens die bewerking uitgevoer word. Dieselfde geld andersom: Daar is slegs een leser, en daarom sal onderbrekings waarin daar 'n *stoot*-bewerking plaasvind, nie die onvoltooide *haal*-bewerking affekteer nie. 'n Implementasie waar die skryfindeks en lengte gestoor word en die leesindeks daarvan afgelei word, sal die verkeerde element gelees word indien die *haal*-bewerking deur 'n *stoot*-bewerking onderbreek word.

Die logiese **EN**-bewerking (&) in Figuur 6.10 werk soos 'n modulus, maar vereis dat die buffergrootte 'n heelgetal van 2 is. Verder moet die grootte van die veranderlike wat die skryfindeks en die leesindeks stoor, ten minste een bis breër wees as die lengte van die buffer, sodat Vergelyking 6.1 geld, ongeag waar die data in die geheue-blok voorkom.

$$\begin{aligned} \text{aantal elemente in buffer} &= (\text{skryfindeks} - \text{leesindeks}) \bmod (2^N) \\ \text{waar } N &= \text{aantal bisse in elke indeksveranderlike} \end{aligned} \quad (6.1)$$

Deur die resultaat van die aftrekking¹⁰ as 'n tekenlose heelgetal (`uintN_t`) te interpreteer, gebeur die modulus-bewerking implisiet. Dit beteken dat die berekening van die bufferlengte in Vergelyking 6.1 atomies is.

¹⁰ Die redelike aanname word gemaak dat 'n "twee-se-komplement" binêre stelsel gebruik word.



Figuur 6.10: Draadbestande bewerkings met die sirkelbuffer. Omdat i) daar slegs een leser en een skrywer is, en ii) die lees- en skryfindeks apart gestoor word, en iii) daar geen vertakkings in hierdie funksies is nie, word die behoefte aan **wedersydse uitsluitings** en atomiese bewerkings omseil. 'n Onderbreking wat plaasvind terwyl een van hierdie funksies uitgevoer word sal nie die werking van die buffer omvergooi nie.

Die aantal elemente bly akkuraat, selfs nadat die buffer oorvol is, of as die beginpunt van die buffer nie by posisie 0 in die geheue-blok begin nie. As die indeksveranderlikes N bisse breed is en die bufferlengte is 2^M , waar $M < N$, kan daar nog $(N - M) \times (2^M)$ elemente na 'n vol buffer gestoot word voordat skryfindeks – leesindeks nie meer 'n akkurate lengte gee nie. Dit beteken egter nie dat daardie waardes nog beskikbaar is nie – slegs die laaste 2^M waardes wat gestoot is, bestaan nog in die geheue-blok. Tabel 6.5 toon voorbeelde waar die lengte van die buffer in verskillende toestande bereken word.

Omdat daar geen ekstra bewerkings of toetse gedoen mag word as elemente na die buffer gestoot of uit die buffer gehaal word nie, is die hoogste bis van die verskil tussen die indeksveranderlikes die enigste manier om tussen 'n leë en 'n vol buffer te onderskei (vergelyk byvoorbeeld rye 1 en 9 in Tabel 6.5). 'n Verdere voordeel van 'n akkurate bufferlengte met 'n oorvol buffer, is dat die leser van die buffer kan weet presies hoeveel elemente verlore gegaan het, mits dit minder as $(N - M) \times (2^M)$ is.

So 'n implementasie is as oop-bron kode beskikbaar (Hylands, 2012). Die kode is vir hierdie projek aangepas om 'n langer buffer-grootte te ondersteun, met **struct**-elemente eerder as **char**-elemente. Die aangepaste bronkode is op DVD E.1 in Bylae E beskikbaar. Alle bewerkings word volledig met **C**-makros geïmplementeer (d.w.s. dit maak nie van funksies gebruik nie), wat beteken dat dit baie vinnig is en gebruik kan word waar **konteksoorskakeling** nie moontlik is nie.

Vir hierdie projek is 'n buffer met $N = 16$ (**16-bis heelgetal sonder teken (uint16_t)**-indekse) en $M = 10$ (1024 elemente) gebruik. Elke element bevat een monster (18 B) met 'n tydstempel (6 B), sodat die buffer in totaal 24 576 B in beslag neem. Met 'n **monstertempo** van 4800 Hz is die buffer dus 213,3 ms lank. Die beperkende faktor vir die lengte van die buffer is die beskikbare geheue op die mikrobeheerder. Met $M = 11$ sou die buffer 2048 elemente kon bevat en 49 152 B in beslag neem, maar die PIC32MX340F512H het slegs 32 kB datageheue.

Opdrag	Skikking				Indekse		Berekende lengte
	00	01	10	11	skryf	lees	
	↓↑ 0	0	0	0	0b000=0	0b000=0	0b000=0 ✓ (leeg)
stoot A	↑ A	↓ 0	0	0	0b001=1	0b000=0	0b001=1 ✓
stoot B	↑ A	B	↓ 0	0	0b010=2	0b000=0	0b010=2 ✓
haal	A	↑ B	↓ 0	0	0b010=2	0b001=1	0b001=1 ✓
stoot C	A	↑ B	C	↓ 0	0b011=3	0b001=1	0b010=2 ✓
stoot D	↓ A	↑ B	C	D	0b100=4	0b001=1	0b011=3 ✓
haal	↓ A	B	↑ C	D	0b100=4	0b010=2	0b010=2 ✓
stoot E	E	↓ B	↑ C	D	0b101=5	0b010=2	0b011=3 ✓
stoot F	E	F	↓↑ C	D	0b110=6	0b010=2	0b100=4 ✓ (vol)
haal	E	F	↓ C	↑ D	0b110=6	0b011=3	0b011=3 ✓
stoot G	E	F	G	↓↑ D	0b111=7	0b011=3	0b100=4 ✓ (vol)
stoot H	↓ E	F	G	↑ H	0b000=0	0b011=3	0b101=5 ✓ (oorvol)
haal	↓↑ E	F	G	H	0b000=0	0b100=4	0b100=4 ✓ (vol)
stoot I	↑ I	↓ F	G	H	0b001=1	0b100=4	0b101=5 ✓ (oorvol)
stoot J	↑ I	J	↓ G	H	0b010=2	0b100=4	0b110=6 ✓ (oorvol)
haal	I	↑ J	↓ G	H	0b010=2	0b101=5	0b101=5 ✓ (oorvol)
stoot K	I	↑ J	K	↓ H	0b011=3	0b101=5	0b110=6 ✓ (oorvol)
stoot L	↓ I	↑ J	K	L	0b100=4	0b101=5	0b111=7 ✓ (oorvol)
stoot M	M	↓↑ J	K	L	0b101=5	0b101=5	0b000=0 ✗
stoot N	M	↑ N	↓ K	L	0b110=6	0b101=5	0b001=1 ✗

Tabel 6.5: Voorbeelde van bewerkinge op 'n sirkelbuffer met plek vir $2^2 = 4$ elemente en indekse wat 3 bisse breed is. Die posises waarna die skryf- en leesindekse wys nadat die opdrag uitgevoer is, word by elke stap onderskeidelik met ↓ en ↑ aangedui. Vanaf opdrag “stoot H” word elemente oorskryf en gaan data verlore. Die aantal elemente bly egter akkuraat tot by opdrag “stoot L”.

6.8 Onderbrekingsprioriteite

Die tyd wat 'n mikrobeheerder¹¹ sal neem om 'n spesifieke stuk kode uit te voer, is deterministies, mits die klokspoed bekend is. Dit is dus 'n ‘intydse’ stelsel. Sodra eksterne onderbrekings egter toegelaat word, word die programuitvoering afhanklik van aktiwiteit op die *intreepoorte* – en dus stogasties.

Onder die aanname dat die uitvoering van 'n sekere deel van die program nie onderbreek sal word nie, is dit wel moontlik om te bereken hoeveel tyd dit sal neem. Presies wanneer die kode uitgevoer sal word en of dit onderbreek sal word, is egter lukraak.

Tabel 6.6 toon die onderskeie *drade* in die bufferprogram, met die onderbrekingsprioriteit vir elkeen.

Die PIC32 laat *vervlegte onderbrekings* toe. 'n Onderbreking met 'n hoër prioriteit kan dus 'n laer-prioriteit onderbrekingshanteringsroetine onderbreek. Dit beteken dat daar (buiten die hooflus) soveel *drade* “tegelyk” op die mikroverwerker kan loop as wat daar *onderbrekingsprioriteitsvlakke* is.

DMA3 en **DMA0** het dieselfde prioriteit, maar **DMA0** het hoër subprioriteit. Indien albei onderbrekings gelyktydig plaasvind, of plaasvind terwyl 'n hoër-prioriteit onderbreking (soos INT1 of INT2) besig is om uit te voer, sal **DMA0** eerste uitgevoer word. As die **DMA0**-onderbreking plaasvind terwyl die **DMA3-onderbrekingsroetine** uitgevoer word, sal dit uitgevoer word sodra die **DMA3-onderbrekingsroetine** klaar is.

Hoë-prioriteit *onderbrekingsroetines* moet kort wees, sodat hulle nie veroorsaak dat *onderbrekingsroetines* met laer prioriteite hul spertye mis nie. Indien dieselfde lae-prioriteit onderbreking

¹¹ Hier word 'n mikrobeheerder sonder 'n bedryfstelsel bedoel. 'n Stelsel soos die **RPI** gebruik 'n skeduleerder wat bepaal wanneer watter programme toegang tot die verwerker kry en is dus ook nie 'n *intydse* stelsel nie.

Vektor	Prioriteit.sub	Frekwensie	Beskrywing
INT2	7.1		Aan die nSS-lyn van die SPI-bus wat met die RPI kommunikeer, gekoppel. Die hanteringsroetine maak die SPI_READY-lyn laag. Die RPI wag dan tot SPI_READY weer aan is voordat dit verder gaan met die transaksie. Omdat die RPI soveel vinniger as die PIC32 is, moet hierdie onderbreking die hoogste prioriteit hê, sodat die SPI_READY-lyn laaggetrek word voordat die RPI begin toets wanneer die lyn hoog word.
INT1	4.0	4800 Hz	Reageer op die nREADY pen van die versyferaar, wat aandui dat 'n nuwe monster gereed is om oor SPI verstuur te word. Begin die DMA-transaksie met die versyferaar. Sien §6.5.
DMA0	3.2		Vind aan die einde van elke DMA-transaksie met die RPI plaas. Dit moet 'n hoër prioriteit as die kommunikasie met die versyferaar hê, want die buffer is in hierdie geval die SPI-slaaf, en die SPI-meester beheer die klok. Die SPI-bus moet daarom so vinnig as moontlik voorberei word vir die volgende transaksie. 'n Vlaggie word gestel om aan te dui dat 'n nuwe opdrag beskikbaar is. Die hooflus toets daarvoor en voer die opdrag uit. Sien §6.6.
DMA3	3.0	4800 Hz	Vind aan die einde van elke DMA-transaksie met die versyferaar plaas. DMA-transaksies met die versyferaar word slegs gebruik om monsters uit die data-register te lees. Sodra 'n monster ontvang is, word 'n vlaggie gestel wat deur die hooflus getoets word. Die hooflus stoot dan die nuwe monster na die buffer. Sien §6.5.
SPI1	2.0		Vind plaas vir elke greep wat oor SPI met die versyferaar gekommunikeer word wanneer DMA nie gebruik word nie. Word slegs gebruik tydens die opstelling van die versyferaar as die mikrobeheerder aangeskakel word. Sien §6.5.
ADC	1.0		Meet die batteryspanning. Sien §6.10.
INT4	Afgeskakel	1 Hz	Die GPS PPS-sein is aan INT4 gekoppel. Elke keer as 'n monster gereed is, word die onderbrekingsvlaggie getoets. Indien 'n PPS inderdaad plaasgevind het, word die tydstempel na die volgende sekonde aangeskuiw en die ranggetal na nul gestel.
(Hooflus)	Dit is 'n oneindige lus wat in die oorblywende tyd tussen onderbrekings uitgevoer word.		Meet die batteryspanning (§6.10). Hanteer toestandsmasjien (§6.11). Hou fouttoestandde dop. Stoot nuwe monsters na die buffer. Hanteer opdragte wat ontvang word.

Tabel 6.6: Die verskillende drade van die bufferprogram, met elkeen se prioriteit, frekwensie en taakbeskrywing.

meer as een keer voorkom terwyl 'n hoër-prioriteit **onderbrekingsroetine** uitgevoer word, is die spertyd gemis, omdat die lae-prioriteit **onderbrekingsroetine** slegs een keer uitgevoer sal word. Vervolgens, as 'n lae-prioriteit onderbreking 'n minimum of konstante frekwensie het, impliseer dit dat alle **onderbrekingsroetines** met hoër prioriteite binne een periode hiervan moet klaarmaak.

Omdat die PIC32 heelwat registers het om te stoor, neem die oorskakeling van een program-konteks na 'n ander langer as met kleiner mikrobeheerders (sien §6.2). Die vertraging tussen die onderbreking en die uitvoering van die hanteringsroetine is dus langer. Die volgende aspekte affekteer hierdie vertraging:

Onderbrekingsprioriteitsvlak Indien prioriteit 7 gebruik word, kan daar geen onderbrekings plaasvind totdat dit klaar is nie. Dit laat die C-kompileerder toe om minder instruksies te gebruik vir die oorskakeling.

Konteksoorskakeling Daar is twee maniere om die programkonteks oor te skakel: Die gewone manier is om al die registers na die programstapel te **stoot**, die nuwe konteks in die registers in te laai en die **onderbrekingsroetine** te begin uitvoer. 'n Vinniger manier is om van skaduregisters gebruik te maak. Dit is 'n ekstra registerbank wat gebruik kan word om hoër-prioriteit **onderbrekingsroetines** direk uit te voer sonder om tyd te spandeer om alles eers na die programstapel te **stoot**.

Funksies Sommige C-kompileerders kan die oorskakeling optimeer wanneer daar geen funksies vanuit die **onderbrekingsroetine** geroep word nie. Dit maak die **onderbrekingsroetine** vinniger wanneer die onderbreking slegs van C-makros gebruik maak, of direk na 'n register skryf, byvoorbeeld om 'n **GPIO**-pen te wip.

Saamsteltaal Indien die roetine eerder heeltemal in **saamsteltaal** geskryf word, is die **konteksoorskakeling** nie nodig nie. Dit bied die heel vinnigste hantering van onderbrekings, maar is ook moeiliker om te implementeer.

Vir INT2 in Tabel 6.6 is **onderbrekingsprioriteitsvlak** 7 gebruik, en 'n enkele C-makro word gebruik om die **SPI_READY**-pen af te skakel, sodat dié onderbreking so vinnig as moontlik hanteer word.

As gevolg van die stogastiese gedrag van 'n stelsel met verskeie onderbrekings, is dit moeilik om te bepaal of die onderbrekingsprioriteite reg opgestel is en die hanteringsroetines vinnig genoeg uitgevoer sal word. Die beste manier om dit te toets, is om die mikrobeheerder eers met die res van die stelsel te integreer. Die program wat in die volgende hoofstuk (§7.5 – 7.7) ontwikkel word, is hiervoor gebruik. Dit is verskeie kere oor 'n periode van meer as 24 h uitgevoer sonder dat enige monsters verlore gegaan het.

6.9 DMA-kanaal-prioriteite

Die vier **DMA**-kanale op die PIC32 kan nie gelyktydig data kopieer nie (Microchip, 2013). Net soos met die onderbrekings, word arbitrasie van die **DMA**-kanale volgens vasgestelde prioriteite gedoen.

DMA0 en **DMA1** word vir **SPI2** gebruik, wat met die **RPI** kommunikeer, terwyl **DMA2** en **DMA3** vir **SPI1** gebruik word. Daar is gekies om **DMA0** en **DMA1** hoër prioriteit te gee as **DMA2** en **DMA3**, omdat kommunikasie met die **RPI** baie vinniger is as met die versyferaar. Indien twee kanale dieselfde prioriteit het, sal die kanaal met die laer nommer eerste sy werk verrig. Tabel 6.7 toon die prioriteit en doel van elke **DMA**-kanaal.

Die **DMA**-prioriteite moet nie met **DMA**-onderbrekingsprioriteite verwar word nie. Die eersgenoemde bepaal watter kanale eerste toegang tot die **DMA**-bus kry en dus eerste die geheue mag manipuleer. Die laasgenoemde bepaal watter onderbrekings eerste hanteer word. Hierdie onderbrekings vind plaas wanneer die **DMA**-kanaal se werk voltooi is.

Kanaal	Prioriteit	Sneller	Aksie
0	3	SPI2 het 'n greep ontvang.	Kopieer greep van SPI2BUF na RAM (lees die greep wat oor SPI2 ontvang is).
1	3	SPI2 het 'n greep verstuur.	Kopieer greep van RAM na SPI2BUF (stuur die volgende greep oor SPI2).
2	0	SPI1 het 'n greep verstuur.	Kopieer greep van RAM na SPI1BUF (stuur die volgende greep oor SPI1).
3	0	SPI1 het 'n greep ontvang.	Kopieer greep van SPI1BUF na RAM (lees die greep wat oor SPI1 ontvang is).

Tabel 6.7: Opstelling van **DMA** vir die bufferprogram.

6.10 Spanningsmonitor

Dit is ongewens om 'n gewone loodsuurbattery heeltemal te ontlaai; dit kan die battery-leeftyd drasties verkort. Wanneer 'n **RPI** skielik afskakel omdat die battery leeg is, is daar 'n kans dat data verlore gaan wat nog nie na die SD-kaart geskryf is nie. Dit is dus nuttig om die battery-spanning te kan meet, sodat die stelsel daarop kan reageer voordat dit te laat is.

Omdat hierdie projek reeds 'n mikrobeheerder het wat aan 'n **RPI** verbind is, was die vinnigste oplossing om 'n eenvoudige spanningsdeler tussen die 5 V-lyn en een van die analoogintreepoorte van die mikrobeheerder te koppel. Die battery-spanning word dan periodies gelees, en indien dit laer is as 'n sekere waarde, word 'n **GPIO**-pen wat aan die **RPI** gekoppel is, hoog gestel. Die mikrobeheerder is geprogrammeer om 'n mate van histerese toe te pas, sodat die **GPIO**-pen nie heen-en-weer skakel wanneer die spanning naby die grens-spanning is nie. Die bronkode hiervoor is op DVD E.1 in Bylae E beskikbaar. Die **RPI** lees hierdie pen periodies en voer 'n veilige afskakel-prosedure uit wanneer 'n battery-waarskuwing plaasvind (sien §7.15).

'n Meer elegante oplossing sou wees om 'n diskrete spanningsvergelyker met histerese te implementeer. Dan hoef die mikrobeheerder nie 'n analoogpoort hiervoor te gebruik of enige verwerkingstyd hieraan te spandeer nie.

6.11 Toestandsmasjien

Die hooflus van die program op die mikrobeheerder onderhou 'n eenvoudige toestandsmasjien, wat deur Figuur 6.11 beskryf word. Daar is drie toestande:

Begin Stel die mikrobeheerder op om te funksioneer soos vroeër in hierdie hoofstuk beskryf:

- Die **SPI**-busse en **DMA**-kanale word opgestel.
- Daar word getoets of die mikrobeheerder se klok reg funksioneer. As die eksterne kristal nie reg werk nie, sal die mikrobeheerder terugval na 'n stadiger, interne klok, wat nie vinnig genoeg is vir die huidige toepassing nie.
- Die sirkelbuffer word opgestel.
- Die interne versyferaar word opgestel.
- Die **MCP3903**-versyferaar word aangeskakel en opgestel.
- Die sagteware word voorberei om opdragte van die **RPI** via **SPI2** te ontvang.

Sodra die **MCP3903** gereed is, skakel die toestandsmasjien oor na die 'normale' toestand. Indien enigiets met die opstelling foutgegaan het, word die fouttoestand betree.

Normaal Hanteer hoër-vlak kommunikasie met die versyferaar en **RPI**:

Foutkode	Betekenis	Waarskynlike oorsaak
0x00	Geen fout.	–
0x01	Antwoord is te lank.	Te veel monsters is op een slag versoek.
0x02	Onbekende opdrag ontvang.	'n Sagtewarefout aan die RPI -kant.
0x04	Fout op SPI bus 1.	SPI1BUF is nie gou genoeg gelees nie.
0x05	Fout op SPI bus 2.	SPI2BUF is nie gou genoeg gelees nie.
0x06	Onmoontlike toestand bereik.	'n Sagtewarefout aan die PIC32 -kant.
0x07	Fout by versyferaar.	Oudiokoppelvlak-bordjie nie aan PIC32 gekoppel nie.
0x08	Fout met klokopstelling.	Stukkende kristal.
0x0B	Buffer leeg.	Meer monsters is versoek as wat beskikbaar is.
0x0C	Twee drade probeer tegelyk stoot .	'n Sagtewarefout aan die PIC32 -kant.
0x0D	Twee drade probeer tegelyk haal .	'n Sagtewarefout aan die PIC32 -kant.
0x0E	Versyferaar nie gereed nie.	Kommunikasie oor SPI1 is te stadig.
0x0F	Kan nie foute korrigeer nie.	Swak verbinding deur SPI2 .

Tabel 6.8: Foutkodes van die sagteware op die **PIC32**. Die 1 B foutkode word herhaaldelik na die **SPI2**-buffer geskryf terwyl die toestandsmasjien in die foutmodus is.

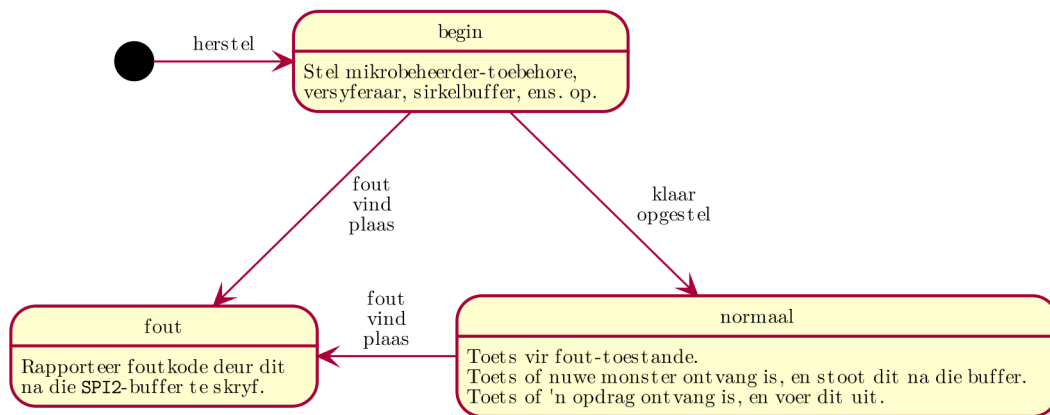
- Toets of 'n fout plaasgevind het. 'n Globale veranderlike word gebruik om die fouttoestand aan te dui. As 'n fout op enige plek in die mikrobeheerder se program voorkom, word hierdie veranderlike na 'n nie-nul waarde gestel, en die toestandsmasjien skakel na die fouttoestand oor.
- Kyk of 'n nuwe monster via **SPI1** van die versyferaar ontvang is. Indien wel, **stoot** dit na die buffer.
- Kyk of 'n nuwe opdrag via **SPI2** van die **RPI** ontvang is. Indien wel, voer dit uit. Die lys van moontlike opdragte word in Tabel 6.4 gelys. Die gebruik van hierdie opdragte deur die **RPI** word in §7.5 behandel.

Fout In die fouttoestand reageer die mikrobeheerder nie meer op opdragte wat deur die **SPI2**-bus ontvang word nie, maar rapporteer 'n foutkode. Al die moontlike foutkodes word in Tabel 6.8 opgesom. Om na die normale werking terug te keer, moet die **RPI** die **PIC32** se **nMCLR**-pen (sien §6.4 en §7.6) vir 'n oomblik laagtrek en dan weer hoogtrek.

Die program kan gestop word deur die mikrobeheerder af te skakel, d.w.s. deur die **nMCLR**-pen (sien §6.4) laag te trek.

6.12 Opsomming

Die ontwerp van 'n digitale buffer vir oudiodata is voltooi. Dit werk deur monsters altyd betyds via **SPI** van die versyferaar te lees en in ewetoeganklike geheue (**RAM**, Random Access Memory) te stoor. Die data word dan via 'n tweede **SPI**-bus aan 'n **RPI** oorgedra, wat in die volgende hoofstuk aandag geniet.



Figuur 6.11: Toestandsmasjien wat deur die hooflus op die mikrobeheerder beheer word.

Hoofstuk 7

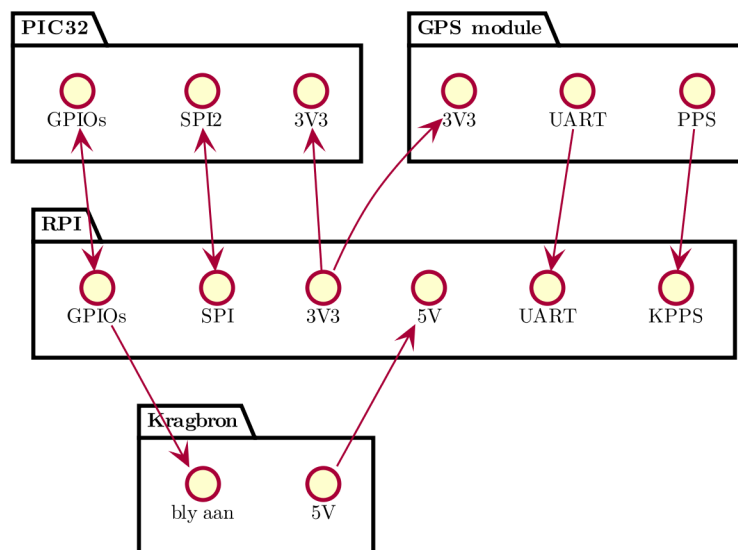
Raspberry Pi

Die **RPI** 3 word in hierdie projek gebruik om data van die buffer-mikrobeheerder te lees, dit te verwerk en dit na 'n **Secure Digital (SD)**-kaart te stoor. Daar is baie oopbron-pakkette vir die **Linux**-bedryfstelsel beskikbaar, wat die sagteware-ontwikkeling makliker maak. Met 1 GB **RAM** en 'n 1,2 GHz vier-kern verwerker is daar genoeg verwerkingskrag om die data te verwerk en nie net te stoor nie. Alhoewel die data in hierdie projek op 'n gewone rekenaar verwerk word nadat die data van die **RPI** af gekopieer is, word daar dus voorsiening gemaak vir dataverwerking op die toestel self in die toekoms. **Linux** is egter nie 'n **intydse** bedryfstelsel nie, wat beteken dat dit uitdagings bied wat nie in **intydse** stelsels soos die program op die **PIC32** voorkom nie.

7.1 Verbindings

Die **RPI** het 'n strook van 2×20 verbindingspenne. Tabel 7.1 toon waarvoor elke pen in hierdie projek gebruik word en in watter afdeling dit bespreek word. Figuur 7.1 bied 'n oorsig oor hierdie verbindings.

Op die **RPI** 2 en **RPI** 3 het **GPIOs** 2 en 3 fisiese optrekweerstande, wat dus nie in sagteware afgeskakel kan word nie ([Raspberry Pi Foundation, 2017e](#)). Al die ander **GPIOs** het op- en aftrekweerstande wat in sagteware gestel kan word. **GPIOs** 2 en 3 kan dus nie gebruik word waar 'n aftrekweerstand nodig is nie (byvoorbeeld die **PPS**-pen – sien §7.13).



Figuur 7.1: Opsomming van verbindings tussen die **RPI** en ander komponente in die stelsel.

§	Verbind met	Doel	RPI-penne		Doel	Verbind met	§
		3,3 V	U — 1	2 — I	5 V	Kragbron	3.2
6.4	PIC nMCLR	MCU-herstel	U — 2 — 3	4 — I	5 V	Kragbron	3.2
3.5	Kragbron	Bly aan	U — 3 — 5	6 —	GND	GND	
		—	— 4 — 7	8 — 14 — U	TXD0	GPS U2RX	7.13
	GND	GND	— 9	10 — 15 — I	RXD0	GPS U2TX	7.13
		—	— 17 — 11	12 — 18 —	—		
		—	— 27 — 13	14 —	GND	GND	
		—	— 22 — 15	16 — 23 — I↑	Afskakel	Skakelaar	7.15
		3,3 V	U — 17	18 — 24 — I↓	PPS	GPS PPS	7.13
6.4	PIC SDI2	MOSI	U — 10 — 19	20 —	GND	GND	
6.4	PIC SDO2	MISO	I — 9 — 21	22 — 25 —	—		
6.4	PIC SCK2	SCLK	U — 11 — 23	24 — 8 — U	CE0#	PIC nSS2	6.4
	GND	GND	— 25	26 — 7 —	—	PIC RE7	6.4
6.4	PIC RE6	—	— 0 — 27	28 — 1 —	—	PIC RE5	6.4
6.4	PIC RE4	—	— 5 — 29	30 —	GND	GND	
6.4	PIC RE3	—	— 6 — 31	32 — 12 —	—	PIC RE2	6.4
6.4	PIC RE1	5 V laag	↓I — 13 — 33	34 —	GND	GND	
6.4	PIC RF1	SPI_READY	↓I — 19 — 35	36 — 16 —	—	PIC RE0	6.4
6.4	PIC RF0	MCU-fout	↑I — 26 — 37	38 — 20 —	—		
	GND	GND	— 39	40 — 21 —	—		

Tabel 7.1: Verbindings volgens die fisiese uitleg van die RPI se penne. Die penne word beskryf met C–B–A | A–B–C waar A die fisiese pennommer is en B die GPIO-nommer is. C dui aan of dit ’n *intree*- of *uittree*poort is. Interne optrek- en aftrekweerstande word onderskeidelik met ↑ en ↓ aangedui. Sommige penne is met die PIC32-mikrobeheerder verbind, maar is slegs vir ontwikkeling gebruik en dien dus nie meer ’n doel nie.

Die RPI sal monsters van die mikrobeheerder via die SPI ontvang en inligting omtrent die huidige tyd en posisie van die GPS-module via die UART ontvang. Die tyd word gebruik om die monsters te sinkroniseer, sodat opnames van verskillende onafhanklike stelsels met mekaar vergelyk kan word om akkurate klankbronopsoring te doen.

7.2 Hardeware-instelling in Linux

’n ‘Device Tree’ (DT) is ’n datastruktuur in ingebedde Linux-stelsels wat die hardeware-opset beskryf, sodat sagtewarepakette reg daarop kan reageer. Die Linux-kern, wat die eerste kode is wat op ’n RPI uitgevoer word, gebruik byvoorbeeld die DT om die RPI se GPIOs op te stel (Raspberry Pi Foundation, 2017b).

Veranderinge aan die DT kan met behulp van ‘Device Tree Overlays’ (DTOs) aangebring word. ’n DTO is ’n konfigurasielêer wat ’n onderdeel van die DT vervang en dus ’n verandering aan die hardewarebeskrywing teweegbring.

Sagteware soos WiringPi deur Henderson (2017) kan gebruik word¹ om poorte te verander as die bedryfstelsel reeds loop, maar die Linux-kern kan dit baie vroeër doen as ’n DTO daarvoor opgestel word. Die volgende pasgemaakte DTO word gebruik om GPIOs 23 en 24 op te stel:

¹ Sien <http://wiringpi.com/the-gpio-utility/> vir voorbeelde.

Riglyn	Beskrywing
<code>dtoverlay=pi3-disable-bt</code>	Skakel die Bluetooth-modem af en laat <code>/dev/ttyAMA0</code> GPIOs 14 en 15 gebruik (Raspberry Pi Foundation, 2017f).
<code>enable_uart=1</code>	Sien §7.3. Impliseer <code>core_freq=250MHz</code> .
<code>dtparam=spi=on</code>	Skakel SPI aan. Sien §7.4. Afhanklik van <code>core_freq=250MHz</code> .
<code>gpu_mem=16</code>	Stel die aantal RAM wat deur die grafikaverwerker gebruik kan word na die minimum toelaatbare waarde van 16 MB (Raspberry Pi Foundation, 2017d), omdat die RPI slegs via SSH gebruik word.
<code>dtoverlay=pps-gpio,</code> <code>gpiopin=24</code>	Laai 'n DTO wat aandui dat GPS PPS aan GPIO24 gekoppel is. Sien §7.13.
<code>dtoverlay=gpio-poweroff,</code> <code>gpiopin=3,</code> <code>active_low=1</code>	Laai 'n DTO wat veroorsaak dat GPIO3 afgeskakel word wanneer die bedryfstelsel stop. GPIO3 is ideaal vir hierdie doel, omdat dit 'n fisiese ingeboude optrekweerstand het.
<code>dtoverlay=grootvoet-gpio</code>	Laai die grootvoet-gpio DTO (in §7.2 beskryf).

Tabel 7.2: 'n Volledige beskrywing van riglyne in `config.txt`.

```

/dts-v1/;
/plugin/;

/ {
    compatible = "brcm,bcm2708";

    fragment@0 {
        target = <&gpio>;
        __overlay__ {
            pinctrl-names = "default";
            pinctrl-0 = <&grootvoet_pins>;

            grootvoet_pins: grootvoet_pins {
                brcm,pins = <24 23>; /* gpio no. */
                brcm,function = <0 0>; /* 0:in, 1:out */
                brcm,pull = <1 2>; /* 2:up 1:down 0:none */
            };
        };
    };
};

```

Hierdie bronkode is op DVD E.1 in Bylae E beskikbaar.

Buiten die instelling van **GPIOs** kan **DTOs** ook gebruik word om randapparatuur soos Bluetooth aan of af te skakel.

Waar 'n gewone rekenaar 'n BIOS of EFI gebruik om die bedryfstelsel af te skop en konfigurasies te hanteer wat deur die bedryfstelsel benodig word sodra dit begin, gebruik die **RPI** 'n tekslêer (Raspberry Pi Foundation, 2017a). Hierdie tekslêer heet `config.txt` en moet op die eerste **partisie** van die **SD**-kaart verskyn. Dit bepaal die verwerker se klokfrekwensie, watter **DTOs** gelaai moet word, ens. Die inhoud van `config.txt` vir hierdie projek word in Tabel 7.2 verklaar.

'n Tweede konfigurasielêer, `cmdline.txt`, bepaal die parameters wat aan die Linux-kern gestuur word wanneer die bedryfstelsel begin word (Raspberry Pi Foundation, 2017c). Tabel 7.3

Kernparameter	Beskrywing
<code>console=tty1</code>	Beveel Linux om die konsole op <code>tty1</code> eerder as op die <code>UART</code> te plaas, omdat dit vir die <code>GPS</code> gebruik gaan word. Sien 7.3 en 7.13.
<code>root=/dev/mmcblk0p2</code>	Gebruik die tweede <code>partisie</code> op die <code>SD</code> -kaart as die <code>lêeroorsprong</code> .
<code>rootfstype=ext4</code>	Die tipe <code>lêergids</code> in die <code>lêeroorsprong-partisie</code> is <code>ext4</code> .
<code>elevator=deadline</code>	Maak seker dat alle in-uit-bewerkings binne 'n eindige tyd klaar maak.
<code>fsck.repair=yes</code>	Probeer die <code>lêergids</code> herstel indien dit beskadig is deurdat die <code>RPI</code> onverwags afgeskakel het.
<code>rootwait</code>	Wag altyd tot die <code>SD</code> -kaart gereed is.
<code>no_hz=off</code>	Skakel kragbesparing af, om gedrag van <code>NTPD</code> te verbeter (Miller en Raymond, 2016). Sien §7.14.

Tabel 7.3: Die Linux-kernparameters in `cmdline.txt`.

beskryf die doel van elke kernparameter.

7.3 UART

'n Serie-poort, oftewel `UART` word benodig om met 'n `GPS`-module te kommunikeer (sien §7.13). Die `RPI` bied twee sulke poorte, wat Linux by `/dev/ttyAMA0` en `/dev/ttyS0` beskikbaar stel (Raspberry Pi Foundation, 2017f).

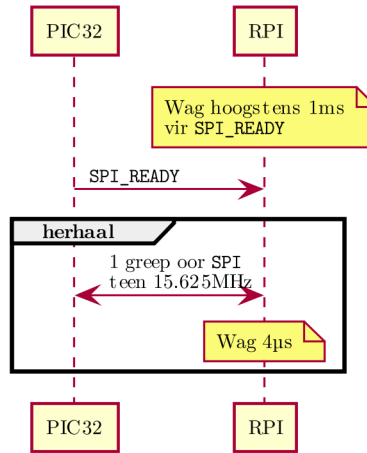
Daar is egter 'n paar uitdagings met betrekking tot die gebruik van hierdie `UARTs`:

- Die bistempo van die `UART /dev/ttyS0` is van die outomatiese frekwensieskalering van die `RPI` se grafika-verwerker (`GPU`, `Graphics Processing Unit`) afhanklik en is dus nie konstant nie. Dit maak hierdie `UART` onversoenbaar met die meeste ander apparate. Die bistempo van `/dev/ttyAMA0` is egter konstant en verstelbaar.
- Die kernfrekwensie van die `RPI 2` se verwerker is 250 MHz, terwyl die `RPI 3` 'n kernfrekwensie van 400 MHz het. Dit het 'n impak op die spoed van die `UART`.
- Die `RPI 3` het 'n Bluetooth-module wat van `/dev/ttyAMA0` gebruik maak.
- Die `RPI 2` verbind `/dev/ttyAMA0` aan penne 8 en 10 (`GPIO` 14 en 15), maar die `RPI 3` verbind `/dev/ttyS0` aan daardie penne.

Met enkele verstellings is dit moontlik om die sagteware vir hierdie projek met beide weergawes van die `RPI` versoenbaar te maak, terwyl die bogenoemde probleme opgelos word:

Bluetooth kan afgeskakel word deur van die `pi3-disable-bt DTO` gebruik te maak. Dit skakel die Bluetooth-hardeware af en ruil die `UARTs` om sodat `/dev/ttyAMA0` weer op penne 8 en 10 (`GPIOs` 14 en 15) beskikbaar is. Die Bluetooth-modemsagteware kan afgeskakel word met die bevel `sudo systemctl disable hciuart`.

Daar is ook 'n konfigurasie-parameter `enable_uart=1` wat die kernfrekwensie van die verwerker na 250 MHz stel, ongeag watter weergawe van die `RPI` gebruik word (Elwell, 2016). Dit impliseer egter dat die Linux-konsole die `UART` sal probeer gebruik. Dit word weer teengewerk deur `console=serial*` van die Linux-kernparameters te verwyder. `console=tty1` word bygevoeg sodat die Linux-konsole steeds deur die hoë definisie `multimediakoppelvlak` (`HDMI`, `High Definition Multimedia Interface`) sigbaar sal wees (Raspberry Pi Foundation, 2017c).



Figuur 7.2: Tydreëling vir kommunikasie tussen die RPI en PIC32.

7.4 SPI

McCauley (2016) het 'n biblioteek van C-drywers vir die BCM 2835-skyfie wat op die RPI gebruik word, geskryf. Daardie biblioteek is in hierdie projek gebruik om die SPI-kommunikasie met die mikrobeheerder (sien §6.6) te implementeer.

Figuur 7.2 toon die opeenvolging van kommunikasie met die PIC32. Geen DMA of onderbrekings word gebruik nie. As die RPI se DMA gebruik word, is die tyd tussen grepe so kort dat die PIC32 se DMA nie vinnig genoeg data na die SPI-buffer kan kopieer nie. Vir elke greep wat oor SPI gestuur word, wag die proses dus totdat die transaksie voltooi is. Dit hou die verwerker baie meer besig as wat nodig is, maar gee fyner beheer oor tyd. Dit is aanvaarbaar omdat daar 4 verwerkerkerne is, min ander prosesse loop en die verwerker baie vinniger is as die PIC32.

As die grepe een vir een gestuur word, kan die tyd tussen grepe gestel word soos nodig. 'n Vertraging van $4\mu\text{s}$ na elke greep is eksperimenteel bepaal as die minimum veilige waarde wanneer die PIC32 teen 80 MHz loop. Die oorhoofse koste van die roep van funksies dra natuurlik effens tot hierdie vertraging by.

In die drywer van McCauley (2016) word die SPI-klokkrekwensie met behulp van 'n klokverdeler gestel. Die kernfrekwensie is reeds vir die UART (§7.3) na 250 MHz gestel. Met 'n klokverdeler van 16 is die SPI-klokkrekwensie dus $250\text{ MHz}/16 = 15,625\text{ MHz}$. 'n Vinniger klok het die kommunikasie onbetroubaar gemaak.

Die maksimum data-deursetting op die SPI-bus kan met behulp van Vergelyking 7.1 bereken word. Die graad van benutting van die bus word deur Vergelyking 7.2 gegee.

$$\text{data tempo} = \frac{1}{\frac{b}{f_{\text{SPI}}} + \tau} < \frac{1}{\frac{8}{15,625\text{ MHz}} + 4\mu\text{s}} \approx 221,6\text{ kWoorde/s} \quad (7.1)$$

$$\text{benutting} = \frac{b}{b + \tau f_{\text{SPI}}} = \frac{8}{8 + 4\mu\text{s} \times 15,625\text{ MHz}} \approx 11,3\% \quad (7.2)$$

waar: b = aantal bisse per woord

f_{SPI} = SPI klokkrekwensie

τ = tyd tussen woorde

Omdat τ groot is met betrekking tot die tyd wat dit duur om een woord te stuur, is die benutting van die bus laag. Siende dat die oudiobuffer teen slegs $115,2\text{ kB} \cdot \text{s}^{-1}$ gevul word, is dit egter vinnig genoeg.

Om te kan verifieer dat data wat oor **SPI** verstuur word korrek is, het die skrywer 'n voorwaartse foutkorreksiekode aan beide kante van die **SPI**-bus geïmplementeer (sien §6.6). Die `ecc_spi_dma`-funksie wat deur Figuur 7.3 beskryf word, hanteer nie net die berekening en dekodeering van die foutkorreksiegrepe nie, maar ook die kommunikasieprotokol wat in 6.4 beskryf is. Die bronkode hiervoor is op DVD E.1 in Bylae E beskikbaar.

Hierdie funksie hou ook 'n stel globale tellers aan om te bepaal hoeveel afgelope transaksies a) korrek was, b) korrigeerbare foute gehad het, of c) onkorrigeerbare foute gehad het, ten einde die kwaliteit van die kommunikasie oor die **SPI**-bus te assesser.

7.5 Hoër-vlak funksies vir kommunikasie met die buffer

Die kommunikasieprotokol met die buffer-mikrobeheerder is reeds in §6.6 uiteengesit. Vir elkeen van die opdragte in Tabel 6.4 is daar 'n hoër-vlak funksie wat die **SPI**-transaksies behartig en die data wat ontvang word, interpreteer.

Funksie `buffer_clear` in Figuur 7.4 herstel die buffer na 'n skoon toestand, sonder om die mikrobeheerder te herstel. Hierdie funksie moet geroep word net voordat die eerste monsters met behulp van `buffer_get_samples` gehaal word, sodat die data wat ontvang word, nuwe data is. As die buffer oorvol geraak het, kan hierdie funksie ook geroep word om dit leeg te maak voordat monsters opnuut gehaal word.

Funksies `buffer_get_time` en `buffer_set_time` in Figure 7.5(a) en 7.5(b) lees en verander die tyd op die mikrobeheerder. Die mikrobeheerder het nie 'n gewone horlosie wat met 'n kristal werk nie, maar hou 'n globale **32-bis heelgetal sonder teken** (`uint32_t`) aan wat met elke **PPS** van die **GPS**-module vermeerder word. Die ranggetal² van die jongste monster word gebruik om die breukdeel van 'n sekonde aan te dui. As die tyd gestel word, word slegs die waarde van die huidige aantal sekondes sedert die Unix-epog gestuur. Wanneer die volgende **PPS** plaasvind, word die ranggetal na nul gestel, en die sekondeteller word vermeerder. Dit maak die tyd op die mikrobeheerder so akkuraat as moontlik (sien §7.14).

Die toestand van die buffer word gelees deur funksie `buffer_get_state` in Figuur 7.6 te roep. Die buffertoestand bestaan uit a) die aantal monsters in die buffer, b) die aantal monsters wat verlore geraak het omdat die kommunikasie tussen die mikrobeheerder en die versyferaar te laat of te stadig plaasgevind het en c) die aantal monsters wat verlore geraak het omdat die buffer oorvol geraak het omdat kommunikasie met die **RPI** te stadig is. Die twee tellers wat die aantal verlore monsters aandui, word herstel as hierdie opdrag uitgevoer word. Hierdie tellers dui dus altyd die aantal verlore monsters sedert die laaste navraag aan.

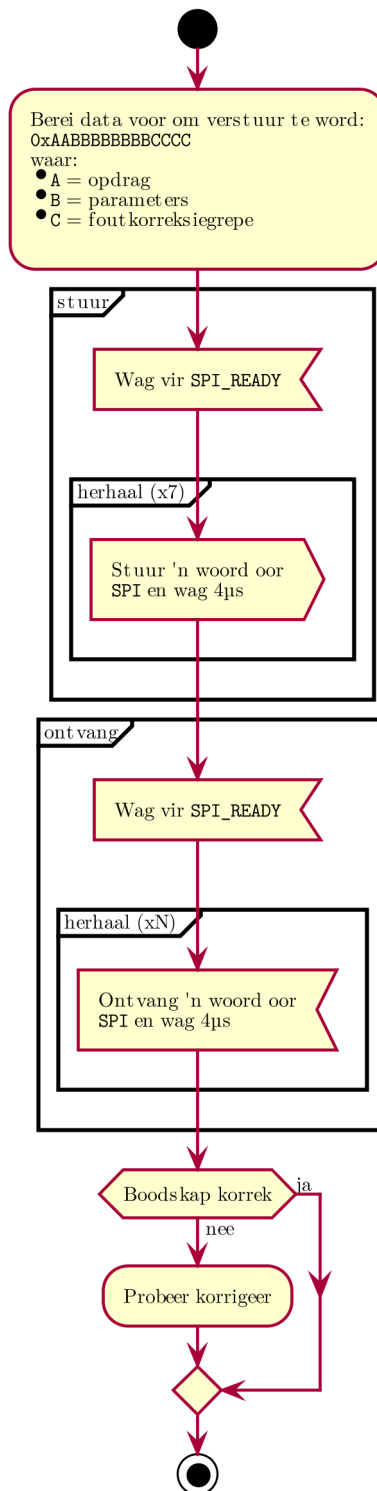
Die belangrikste funksie is `buffer_get_samples` in Figuur 7.7, wat 'n gegewe aantal monsters probeer haal. Elke monster bestaan uit 24 B volgens die struktuur in Tabel 7.4. Nadat die monsters na die **RPI** oorgedra is, word dit in 'n **C struct** gestoor. As gevolg van hoe **C** die inhoud van 'n **struct** in geheue inpak, neem elkeen van die veranderlikes in Tabel 7.4 4 B in beslag. Elke monster neem dus 32 B op die **RPI** in beslag.

In 'n stelsel met greep-geadresseerde geheue neem 'n getal wat groter as een greep is, soos 'n `uint32_t`, meer as een adres in beslag. Die getal word dus in grepe opgebreek as dit in die geheue gestoor word. Sommige stelsels stoor die mees beduidende greep by die laagste adres, oftewel 'groot kant eerste', en ander stoor die mins beduidende greep by die laagste adres, oftewel 'klein kant eerste'. Daar word dus gepraat van die '**kantigheid**'³ van die voorstelling van 'n getal in geheue.

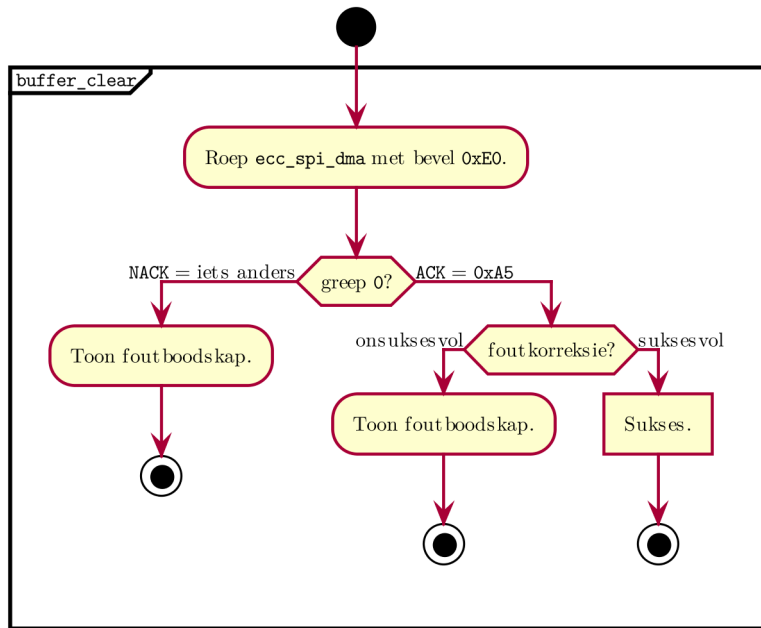
In die meeste implementasies van **C** is daar nie 'n **24-bis heelgetal met teken** (`int24_t`) tipe nie. Om 'n skikking `x` van 3 grepe as 'n **grootkantige** `int24_t` `a` te interpreteer, word die grepe

²n Ranggetal is 'n aanduiding van volgorde. Elke monster kry 'n ranggetal wat aandui hoeveel monsters al in daardie sekonde ontvang is.

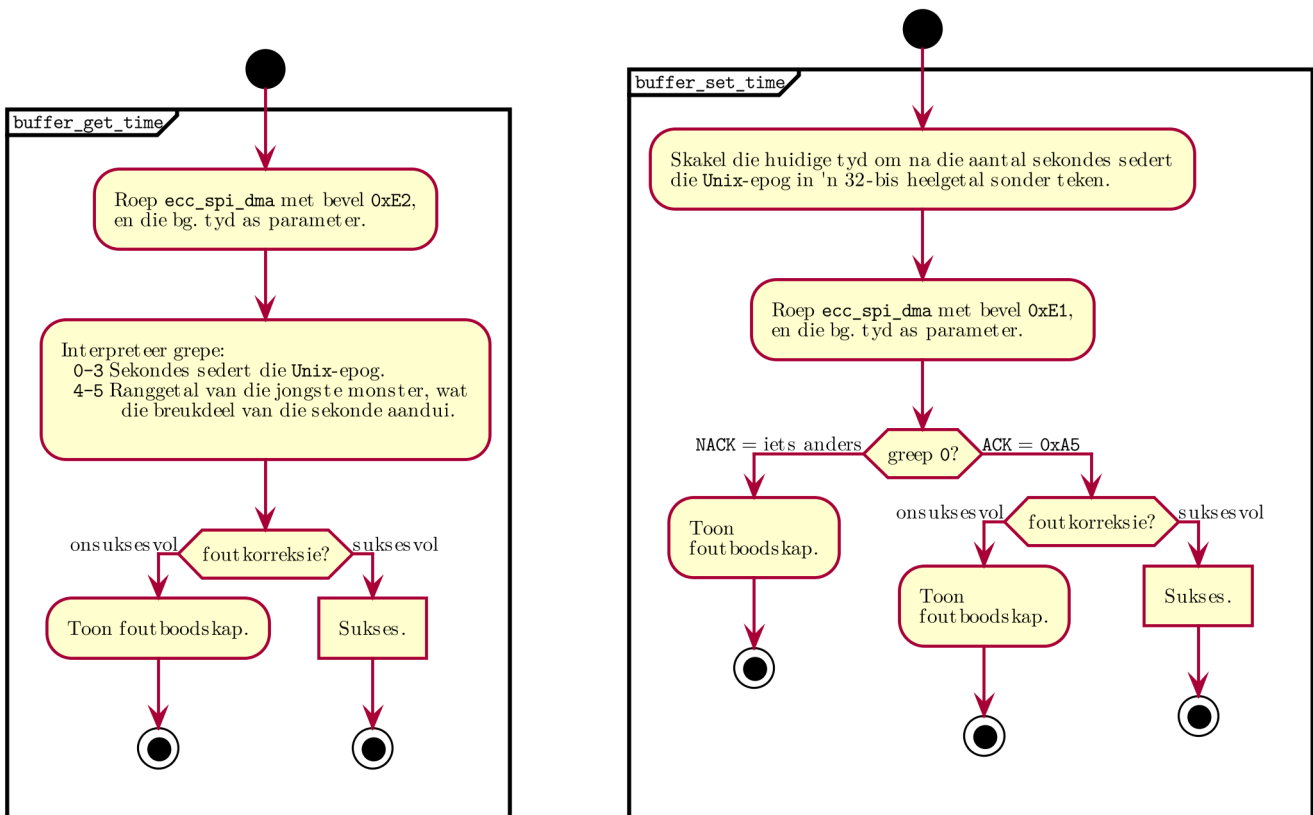
³ Die term **kantigheid** vind sy oorsprong in 'n satire deur Swift (1726), waarin die mense van Lilliput en Blefuscu in 'n stryd gewikkel is oor 'n lewensbelangrike vraag: *Aan watter kant behoort jy 'n eier oop te kap?* In die een-en-twintigste eeu kwel dieselfde moeilike vraagstuk duisende ingenieursstudente wêreldwyd.



Figuur 7.3: Die `ecc_spi_dma`-funksie hanteer kommunikasie met die PIC32 volgens die kommunikasieprotokol wat in 6.4 beskryf is. Dit hanteer ook die kodering en dekodeering van foutkorreksiegrepe en versamel statistiek omtrent die kwaliteit van boodskappe wat gestuur en ontvang word. Fout hantering word ter wille van bondigheid by hierdie diagramme uitgelaat.



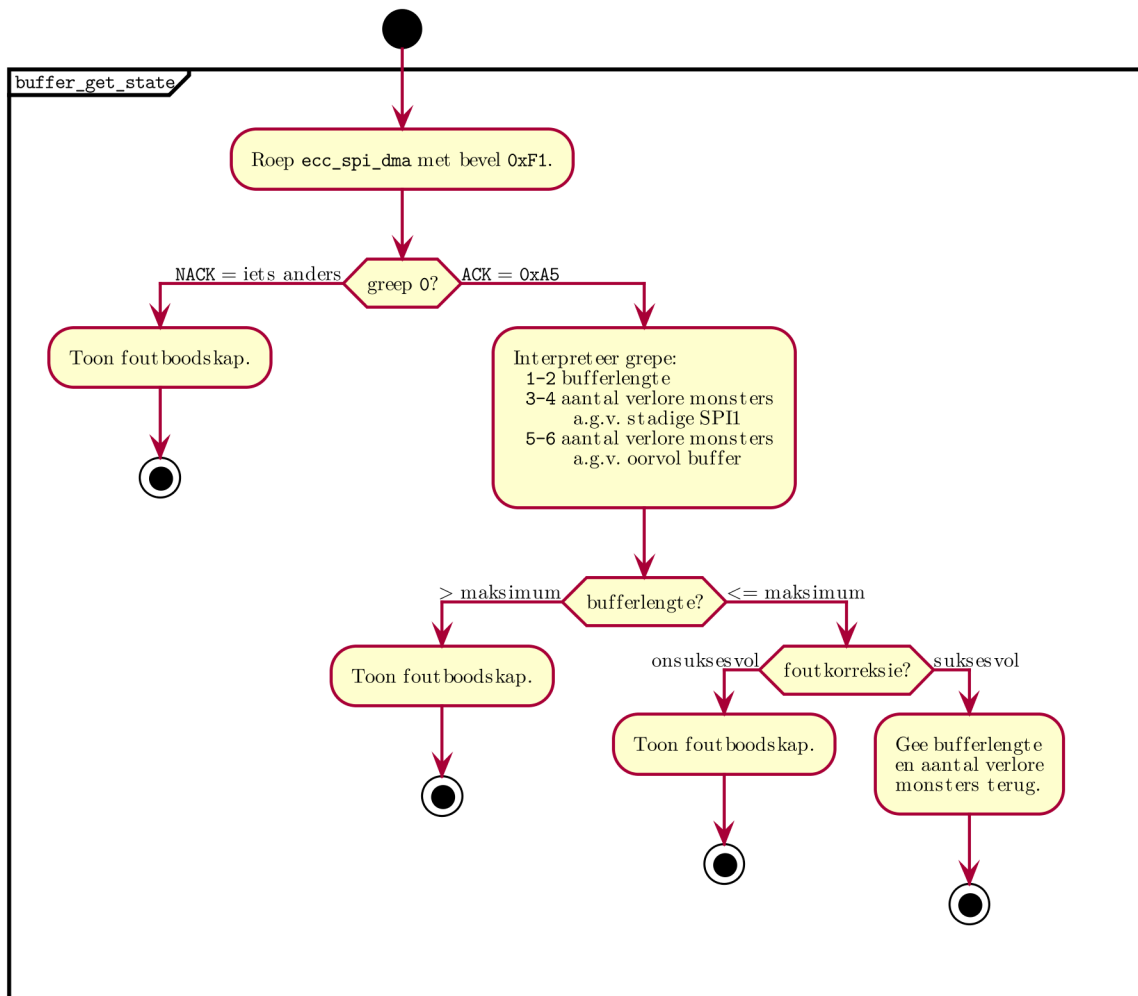
Figuur 7.4: Funksie `buffer_clear` herstel die buffer na 'n skoon toestand, sonder om die mikrobeheerder te herstel.



(a) Funksie `buffer_get_time` lees die tyd van die mikrobeheerder. Die aantal sekondes sedert die Unix-epog dui die sekondes aan, en die ranggetal van die jongste monster dui die breukdeel van 'n sekonde aan.

(b) Funksie `buffer_set_time` skryf die huidige aantal sekondes sedert die Unix-epog na die mikrobeheerder. Die breukdeel van 'n sekonde word gekorrigeer sodra die mikrobeheerder op die volgende PPS van die GPS reageer.

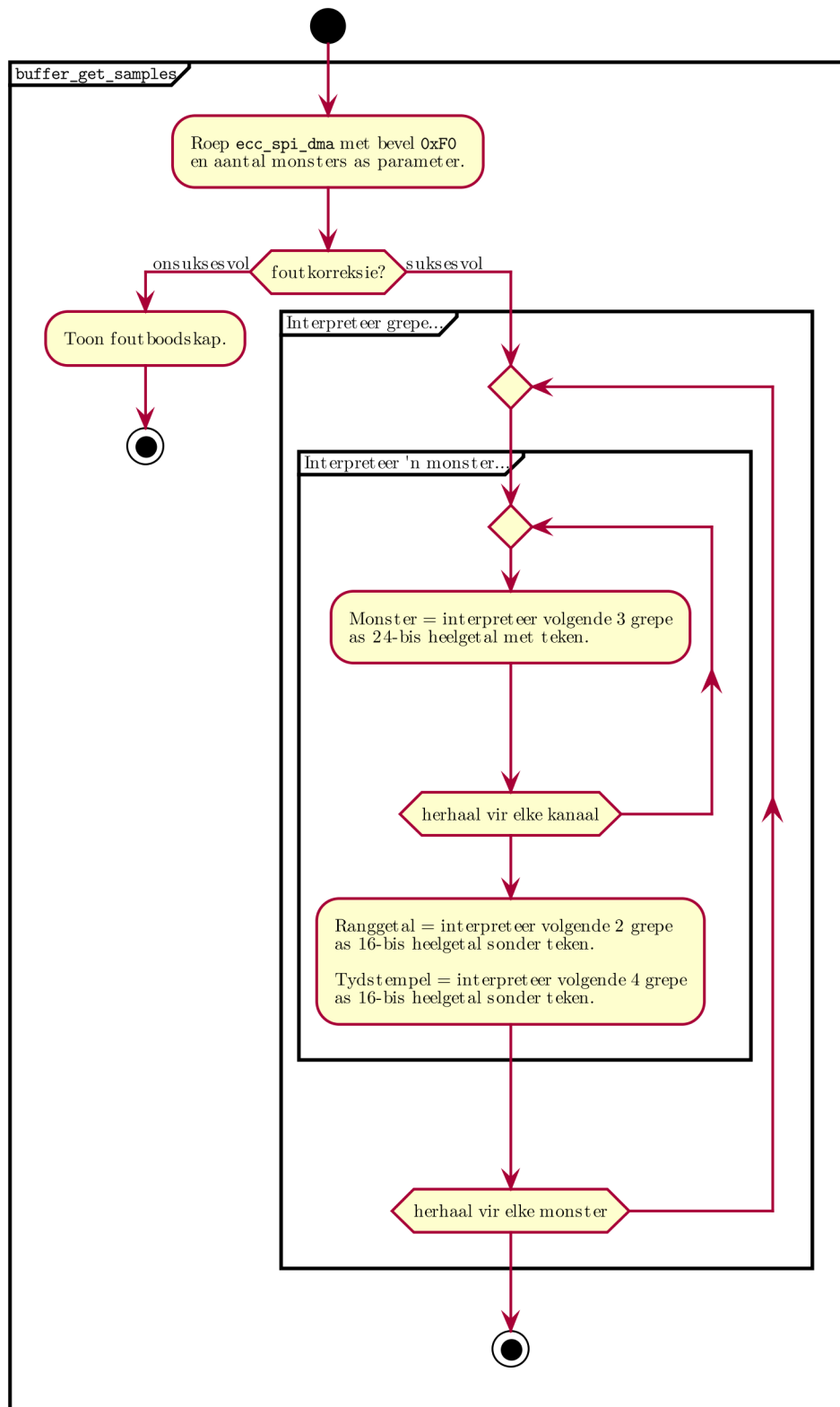
Figuur 7.5: Funksies om die mikroverwerker se tyd te lees en stel.



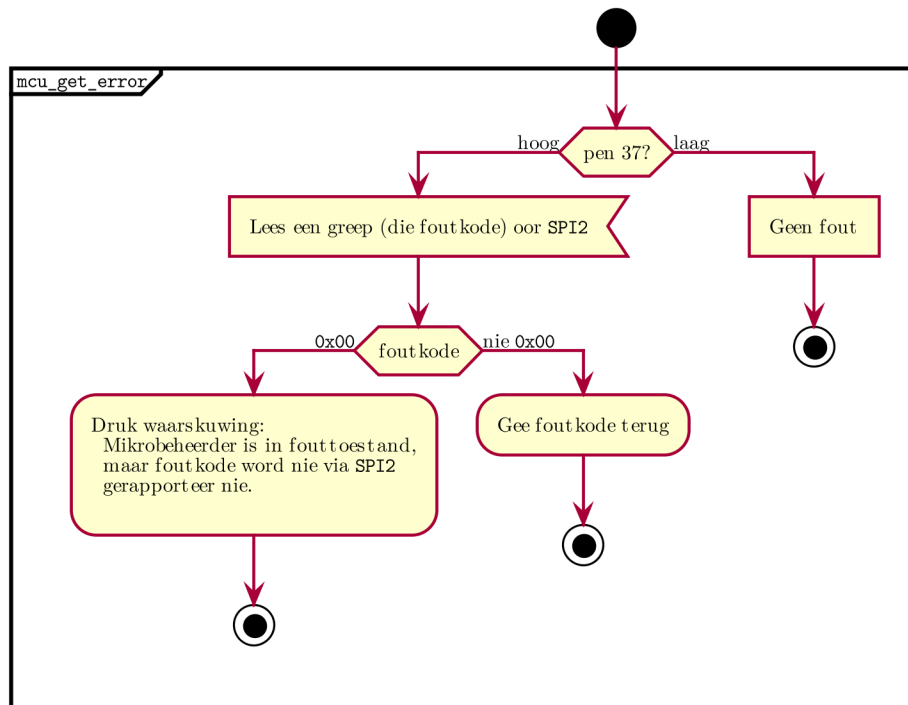
Figuur 7.6: Funksie `buffer_get_state` verkry die toestand van die buffer. Dit veroorsaak ook dat die mikrobeheerder die tellers van die aantal verlore monsters herstel.

Grepe	Inhoud	Tipe	Kantigheid
0–2	Kanaal 1	<code>int24_t</code>	Groot
3–5	Kanaal 2	<code>int24_t</code>	Groot
6–8	Kanaal 3	<code>int24_t</code>	Groot
9–11	Kanaal 4	<code>int24_t</code>	Groot
12–14	Kanaal 5	<code>int24_t</code>	Groot
15–17	Kanaal 6	<code>int24_t</code>	Groot
18–19	Ranggetal	<code>uint16_t</code>	Klein
20–23	Tydstempel	<code>uint32_t</code>	Klein

Tabel 7.4: Die struktuur van elke monster wat via **SPI** ontvang word.



Figuur 7.7: Funksie `buffer_get_samples` probeer 'n gegewe aantal monsters haal en interpreteer.



Figuur 7.8: Funksie `mcu_get_error` toets die fouttoestand van die mikrobeheerder.

van hoog na laag in 'n 32-bis heelgetal met teken (`int32_t`) ingepak, waarna dit 8 bisse na regs verskuif word. Die verskuiwing na regs sorg dat die teken behoue bly:

```
int32_t monster = (x[0] << 24 | x[1] << 16 | x[2] << 8) >> 8;
```

Hierdie oplossing werk, ongeag van die kantigheid van die stelsel wat die bostaande C-kode uitvoer. Die interpretasie van 'n kleinkantige `uint16_t` of `uint32_t` is eenvoudiger:

```
uint16_t ranggetal = y[1] << 8 | y[0];
uint32_t tydstempel = z[3] << 24 | z[2] << 16 | z[1] << 8 | z[0];
```

7.6 Hantering van fouttoestande

Indien iets foutgaan, sal die mikrobeheerder se toestandsmasjien (sien §6.11) 'n fouttoestand betree, waarin 'n foutkode op die SPI bus gerapporteer sal word. Al die moontlike foutkodes is reeds in Tabel 6.8 gelys.

Figuur 7.8 toon hoe die fouttoestand verkry word. Pen 37 (GPIO26) word gelees om te sien of die mikrobeheerder in 'n fouttoestand is. Indien wel, word 'n greep oor SPI verstuur om die foutkode te lees. Hierdie foutkode is handig vir ontwikkelingsdoeleindes, maar sou ook gebruik kon word om die fouttoestand te probeer beredder sonder om die mikrobeheerder te herstel (en dus monsters te verloor). Dit is egter nog nie geïmplementeer nie – die mikroverwerker word altyd herstel as daar probleme is (sien §7.7).

7.7 Program wat monsters haal

Die funksies in §7.4, §7.5 en §7.6 word gebruik om een program in C te skryf, wat monsters via SPI uit die buffer haal en dit na standaarduitree (`stdout`, standard output) skryf. Die program heet `app_stream_all_channels`, en die bronkode is op DVD E.1 in Bylae E beskikbaar. Hierdie

program bestaan hoofsaaklik uit drie funksies wat deur Figure 7.9, 7.10 en 7.11 beskryf word. Daar is ook voorsiening gemaak om die program met CTRL+ ('SIGQUIT') te stop, maar daardie funksionaliteit word ter wille van bondigheid uitgelaat. Op 'n soortgelyke manier word CTRL+C gebruik om vir die program te sê om statistieke na die loglêer te skryf.

Hierdie program werk, maar in die vroeë weergawes was daar baie probleme met die buffer wat oorstroom. Die volgende paar afdelings beskryf hoe die probleem opgelos is.

7.8 Isolاسie van 'n verwerkerkern

Omdat daar meer as een verwerkerkern is, kan die program wat met die versyferaar kommunikeer op 'n geïsoleerde kern loop. Om dit te doen, word die `isolcpus=1` Linux-kernparameter by die bevelreël in `/boot/cmdline.txt` gevoeg (Torvalds *et al.*, 2017a). Dit veroorsaak dat geen prosesse deur die skeduleerder aan die tweede verwerkerkern (CPU 1) toegeken sal word nie en dat onderbrekingsversoeke slegs deur die ander verwerkers hanteer sal word. Die program word dan met behulp van die `taskset`-bevel aangesê om op die tweede verwerkerkern te loop.

Om die program op 'n aparte verwerkerkern te laat loop, het egter die situasie vererger. Daar is nie verdere ondersoek ingestel na hoekom dit so is nie, omdat proses-skedulering, wat in die volgende afdeling bespreek word, 'n voldoende oplossing gebied het.

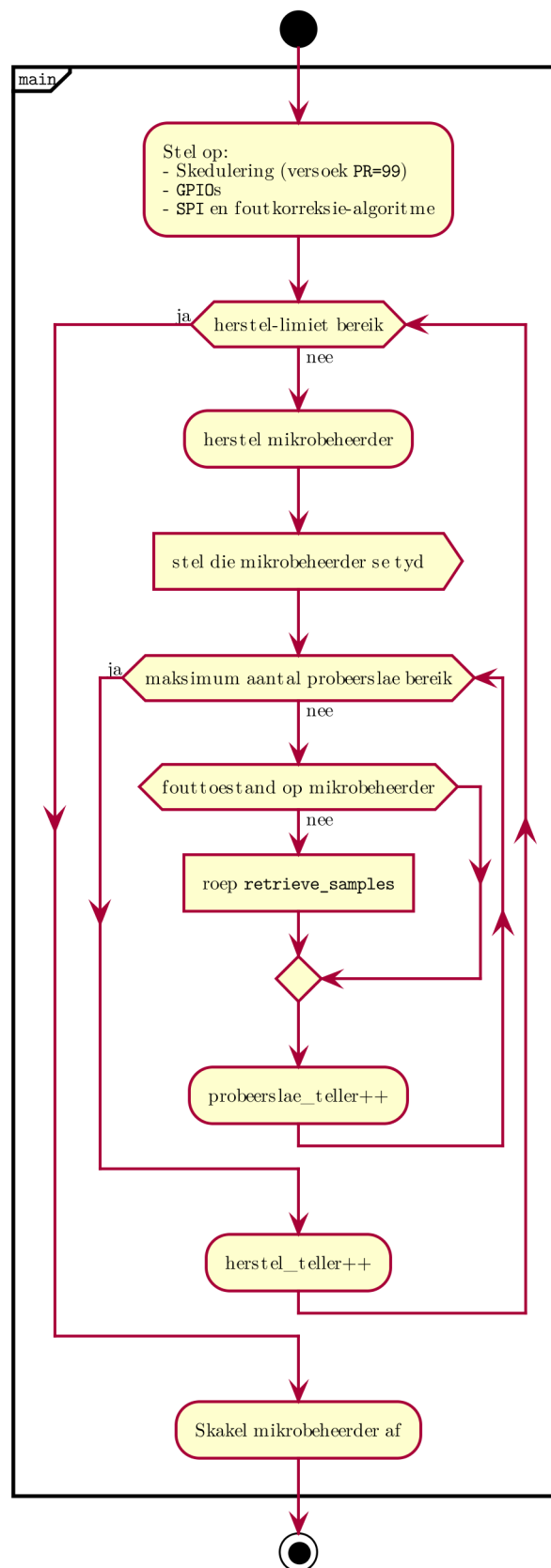
7.9 Skedulering van prosesse

Prosesse op Linux het 'n sekere **beleefdheid** (**NI**, **Niceness**) teenoor ander prosesse, wat bepaal hoeveel tyd die proses graag wil hê in vergelyking met ander prosesse. Dit moet nie met die **prioriteit** (**PR**) van 'n proses verwar word nie. **NI** word gebruik om **PR** te bereken, wat op sy beurt gebruik word om te bepaal hoe gereeld die skeduleerder hierdie proses toegang tot 'n verwerker sal gee. **NI** kan met behulp van die `nice`-bevel (vir nuwe prosesse) of `renice`-bevel (vir bestaande prosesse) verander word. Beleefde prosesse (met 'n hoë **NI**-waarde) versoek 'n lae **PR** en gee dus meer geredelik aan ander prosesse kans om uitgevoer te word. Vir die mees onbeleefde prosesse is $NI = -20$.

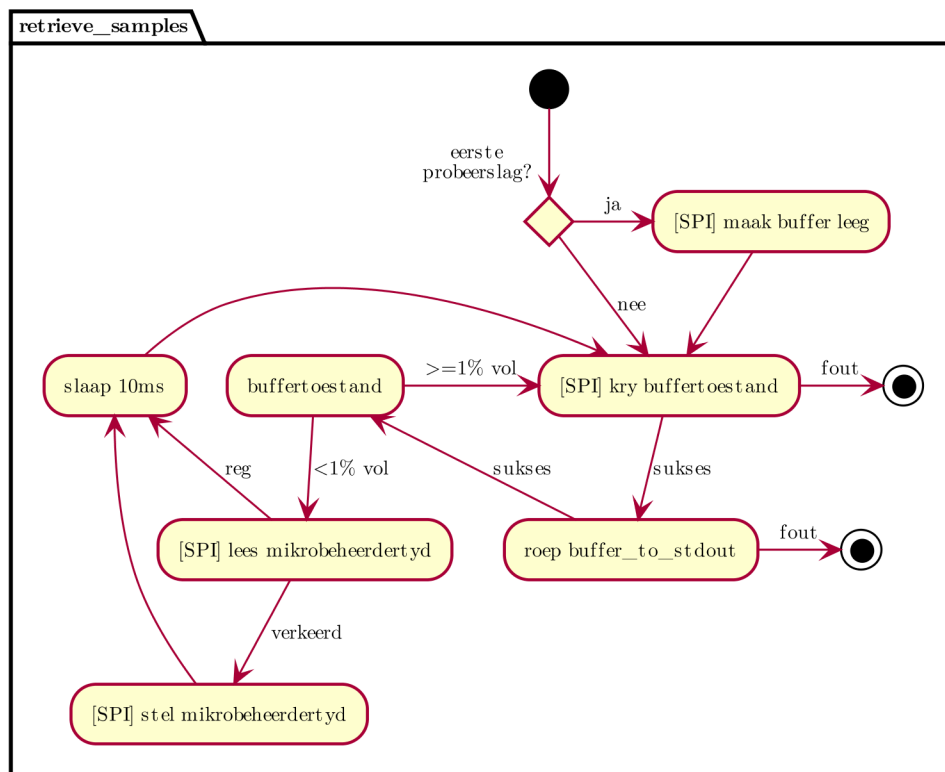
'n Ander manier om 'n hoë prioriteit te verkry, is om **supergebruikersregte** (**sudo**) te gebruik om **PR** na die hoogste moontlike waarde (99) te stel. Prosesse met $PR = 99$ word as 'intyds' bestempel, omdat dit voorrang neem bo alle ander prosesse. Indien so 'n proses al die verwerkingstyd gebruik, sal die bedryfstelsel nie aan enige ander prosesse kan aandag gee nie.

Buiten 'n prioriteit, het elke proses ook 'n **skeduleringsbeleid**. Die verskillende beleide word in twee kategorieë opgedeel:

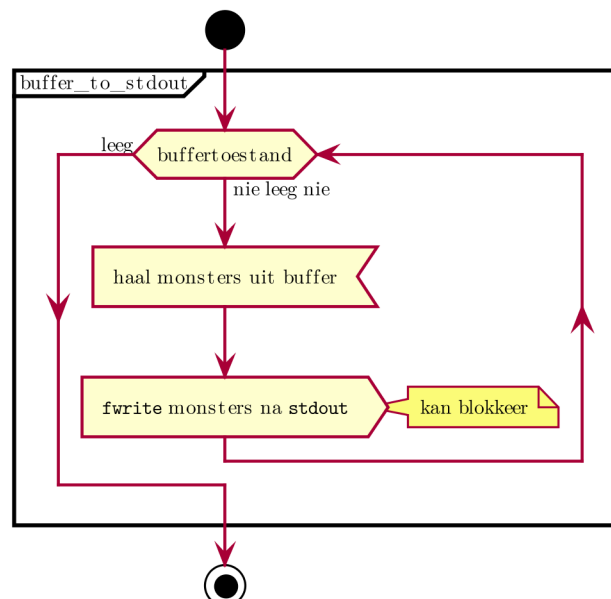
- Normale **skeduleringsbeleide**. Die prioriteit van prosesse met een van hierdie **skeduleringsbeleide** word geïgnoreer.
 - **SCHED_OTHER** is die verstek **rondomtalie**-beleid waar elke proses om die beurt toegang tot 'n verwerker kry.
 - **SCHED_BATCH** is soortgelyk aan **SCHED_OTHER**, maar is bedoel vir nie-interaktiewe prosesse, en die beurte is heelwat langer.
 - **SCHED_IDLE** is bedoel vir agtergrondtake met 'n baie lae prioriteit.
- **Intydske skeduleringsbeleide**. Hierdie beleide neem die prioriteit van elke proses in ag, sodat die hoeveelheid tyd wat aan elke proses spandeer word, beheer kan word. Prosesse met hoër prioriteite word altyd voor prosesse met laer prioriteite uitgevoer. As 'n proses met 'n hoër prioriteit wakker word, sal 'n laer-prioriteit proses so gou as moontlik onderbreek word sodat 'n verwerker aan die hoë-prioriteit proses toegeken kan word. Die uitvoering van prosesse met dieselfde prioriteit word deur die **skeduleringsbeleid** bepaal.



Figuur 7.9: Die hooflus van die program roep die `retrieve_samples`-funksie herhaaldelik en herstel die mikroverwerker as 'n fouttoestand voorkom.



Figuur 7.10: Die `retrieve_samples`-funksie sal aanhou om `buffer_to_stdout` te roep, so lank as wat daar nog monsters in die buffer is. As die buffer minder as 1% vol is, kan daar aan ander sake aandag gegee word, soos om die mikroverwerker se tyd reg te stel, indien nodig.



Figuur 7.11: Die `buffer_to_stdout`-funksie haal soveel as moontlik monsters per `SPI`-transaksie. Dit word herhaal totdat soveel monsters gehaal is as wat in die buffer was toe die buffertoestand laas gelees is (in `retrieve_samples`, Figuur 7.10). Die `fwrite`-funksie kan die uitvoering van die program lank blokkeer indien `stdout` direk aan 'n fisiese stoormedium soos 'n `SD`-kaart gekoppel is. Hierdie probleem word in §7.10 opgelos.

- `SCHED_FIFO` is 'n beleid waar die eerste proses van 'n gegewe prioriteit wat wakker word, voorrang het bo prosesse met dieselfde prioriteit wat later wakker word.
- `SCHED_RR` is 'n **rondomtalie**-beleid, waar elke proses slegs 'n sekere tyd het om te loop, voordat dit deur 'n ander proses van dieselfde prioriteit onderbreek sal word.

Die program versoek 'n prioriteit van 99 en stel die skeduleerder na **FIFO** met behulp van die `sched_setscheduler`-funksie in `sched.h` (Torvalds *et al.*, 2017c). Dit is dus nie nodig om `nice` of `renice` te gebruik nie, maar die program moet met `sudo` begin word. Hierdie hoë prioriteit beteken dat die program voorrang neem bo alle ander prosesse op die bedryfstelsel en dus 'n groter waarskynlikheid het om alle data betyds uit die monsterbuffer te lees. 'n Tweede, lae-prioriteit draad word geskep om foutboodskappe en statusboodskappe na 'n loglêer te skryf.

Omdat daar altyd ander bedryfstelselkomponente is wat ook met 'n prioriteit van 99 loop, is dit nodig om iewers in die program die funksie `sleep` of `sched_yield` te roep. Hierdie funksies staan die gebruik van die verwerker aan ander prosesse af totdat die skeduleerder besluit om die program weer 'n beurt te gee. As die proses op die verkeerde tyd gaan slaap, en die skeduleerder maak dit nie betyds wakker nie, sal die buffer oorstroom. As hierdie tegniek egter met oorleg gebruik word, kan dit die **intydse** gedrag van die program drasties verbeter.

In Figuur 7.10 word gebruik van die verwerker slegs afgestaan as die buffer minder as 1 % vol is, sodat ander prosesse juis sal loop wanneer hierdie proses dit kan bekostig. Dit verlaag die kans dat hierdie program later tydens die uitvoering van 'n tyd-kritiese stuk kode – soos 'n **SPI**-transaksie – onderbreek sal word.

'n Sekondêre rede vir die gebruik van `sleep` of `sched_yield` is sodat die verwerkerkern waarop die proses uitgevoer word nie 100 % verbruik sal toon nie. Opsigself is dit egter nie 'n groot probleem op 'n multi-kern rekenaar soos die **RPI** nie, maar minder verwerking veroorsaak wel 'n langer batteryleeftyd. Wanneer die `app_stream_all_channels` program loop, staan die verwerkergebruik op een kern by ongeveer 60 %, met amper geen verbruik op die ander kerne nie. Daar is dus nog baie beskikbare verwerkingstyd wat in die toekoms gebruik kan word om die data op die toestel self te verwerk.

Die parameter wat aan die `sleep`-funksie gegee word (10 ms in Figuur 7.10) is die minimum tydsduur waarvoor beheer afgestaan word. Die maksimum tydsduur is van verskeie faktore afhanklik, insluitend *a*) die prioriteit van die proses, *b*) die skeduleerderbeleid, *c*) en hoeveel tyd ander prosesse neem.. As die skeduleerder nie aan hierdie program aandag gee voordat die buffer volraak nie, sal monsters verlore gaan.

Wagner *et al.* (2014) gee die volgende raad vir die ontwikkeling van **intydse** programme:

- Moenie programme met prioriteit 99 uitvoer nie. Hierdie raad word doelbewus geïgnoreer omdat die bufferlengte so kort is. Omdat daar seker gemaak word dat die proses slaap wanneer dit kan, sal die verwerkerkern nie 100 % benut word nie en dus wel tyd hê om aan bedryfstelselkomponente aandag te skenk.
- Moenie in 'n lus wag vir iets om te gebeur nie. Dit is egter presies wat vir kort tydjies gedoen moet word om tydens **SPI**-transaksies vinnig op vallende of stygende **GPIO**-penne soos `SPI_READY` te kan reageer (sien §7.4). Dit is moontlik om 'n onderbreking te registreer wat die proses sal wakker maak as 'n **GPIO**-pen styg, maar die oorhoofse koste hiervan is te groot en sou veroorsaak dat die tydsberekening van die **SPI**-kommunikasie nie meer deterministies is nie.
- Hou bevele soos `fwrite`, wat **insette** en **uitsette** hanteer, weg van kode wat aan **intydse** vereistes moet voldoen. Die effek van `fwrite` en die oplossing vir hierdie probleem is reeds in §7.7 bespreek. §7.10 bespreek die hantering van in-uit-strome in meer detail.

7.10 Hantering van in-uit-strome

Die `fwrite`-funksie in C gee data aan die Linux-kern oor, wat dit sal buffer voordat dit fisies na die SD-kaart uitgeskryf word. As die program se `stdout` na 'n lêer gestuur word en die Linux-kern se buffers raak vol, sal die `fwrite`-funksie blokkeer terwyl data fisies van die buffers na die fisiese medium geskryf word (Pornin, 2010). Dit kan heelwat langer neem as die lengte van die buffer, wat veroorsaak dat monsters verlore sal gaan.

Hierdie probleem word opgelos deur die `stdout`-stroom na 'n buffer te stuur wat in die RPI se RAM onderhou word. Vandaar word die buffer deur 'n lae-prioriteit proses gelees, wat die data na die SD-kaart skryf. Die Pipe Viewer (`pv`) program deur Wood (2017) kan hiervoor gebruik word:

```
sudo ./app_stream_all_channels | pv --quiet --no-splice -B 10m > opname.bin
```

- `--quiet` keer dat `pv` enigiets na standaardfoutuittree (`stderr`, standard error output) skryf.
- `splice` is 'n optimering in `pv` waar standaardintree (`stdin`, standard input) direk aan `stdout` gekoppel word, sonder 'n buffer tussenin. `--no-splice` skakel hierdie optimering af.
- `-B 10m` stel die buffergrootte na 10 MB, wat 91 s se monsters kan stoor. Dit behoort oorgenoeg tyd te gee om seker te maak dat die data na die SD-kaart geskryf kan word.

Die enigste nadeel hieraan is dat die inhoud van die RAM-buffer verlore sal gaan as die RPI onverwags sou afskakel. Hierdie probleem word egter opgelos deur die program te stop wanneer die battery swak is, maar voordat die RPI afskakel (sien §7.15).

7.11 Geheue-bestuur

Skikkings en globale veranderlikes neem nie dieselfde geheue-area in beslag as die program waarin dit gedefinieer word nie. Wanneer die proses begin, word virtuele geheue daarvoor toegeken. Die virtuele geheue kan egter óf in RAM, óf in ruilgeheue (d.w.s. op die SD-kaart) voorkom. As die program hierdie geheue wil lees of skryf, vind daar 'n geheue-uitruiling tussen die RAM en SD-kaart plaas.

As dit gebeur, word die program vir lang tye onderbreek en is die RAM-buffer wat in §7.10 bespreek is, van nul en gener waarde. Om dit te voorkom, word die `mlockall`-funksie geroep om seker te maak dat alle veranderlikes van die huidige proses na RAM verskuif word (Drepper *et al.*, 2017). Dit word gedoen voordat kommunikasie met die PIC32 begin.

Daar word aangeneem dat die buffer wat deur `pv` geskep word, altyd in RAM sal wees.

Om heeltemal seker te wees dat die program nie deur uitruiling onderbreek sal word nie, word die `ruilgeheue` van die bedryfstelsel heeltemal afgeskakel, sodat alle virtuele geheue in RAM moet voorkom. Die roep van die `mlockall`-funksie word behou, omdat dit 'n foutkode teruggee wanneer dit nie moontlik is om al die veranderlikes na RAM te skuif nie. In daardie geval word die program met 'n foutboodskap gestaak. Dit dien dus as versekering dat die veranderlikes in RAM is wanneer die program uitvoer.

7.12 SD-kaart

Die SD-kaart berg nie net die bedryfstelsel van die RPI nie, maar ook die oudiodata wat opgeneem word. Die bedryfstelsel en programme neem ongeveer 2,2 GB in beslag, wat op 'n 16 GB kaart ongeveer 12,7 GB vir opnames laat⁴.

⁴ 'n Nuwe "16 GB SD-kaart" behoort ongeveer 16×10^9 adresseerbare grepe te hê, wat eintlik 14,9 GB is, omdat grepe in veelvoude van 1024 eerder as 1000 gemeet moet word. Sommige SD-kaarte het nóg minder adresseerbare grepe, omdat 'n paar miljoen bisse gereserveer word om in te staan vir ander grepe wat breek.

Om ruimte op die SD-kaart te spaar, kan 'n sameperser soos **pigz**⁵ deur Adler (2016) gebruik word.

```
sudo ./app_stream_all_channels | pv --quiet --no-splice -B 10m | \
    pigz -c > opname.bin.gz
```

In hierdie voorbeeld is **app_stream_all_channels** die enigste proses wat met 'n hoë prioriteit loop. Die **pigz**-program maak dus van die oorblywende verwerkertyd gebruik. 'n Gunstige newe-effek van die gebruik van samepersing, is dat daar minder gereeld na die SD-kaart geskryf word.

Wanneer die SD-kaart vol is, sal **pigz** **ineenstort**, wat veroorsaak dat **pv** en **app_stream_all_channels** die SIGPIPE sein kry. Die lêer **opname.bin.gz** kan egter steeds met **zcat** gelees word, en slegs die laaste monter wat na die kaart geskryf is, is moontlik korrupt.

In een toets het die bg. opdrag 134 956,183 s (ongeveer 37,5 h) geneem om 12 304 564 224 B (ongeveer 11,5 GB) te skryf voordat die SD-kaart volgeraak het. Die program rapporteer dat 647 823 865 monsters versamel is, waarvan geeneen tussen die PIC32 en die RPI verlore gegaan het nie, en dat 0,000 030 % van alle SPI-transaksies korrigeerbare foute gehad het. 647 823 865 monsters teen 4800 Hz sou egter 134 963,305 s neem, wat beteken dat die laaste 7,12 s se audio (ongeveer 34 187 monsters) nog in die RAM-buffer was toe die SD-kaart vol raak. Daar behoort dus ongeveer 647 789 678 monsters in die gestoorde lêer te wees, wat sonder samepersing elk 32 B in beslag neem (sien §7.5), vir 'n totaal van 20 729 269 696 B (ongeveer 19,3 GB). Die grootte van **opname.bin** nadat dit met **zcat** opgeblaas is, is 20 730 119 561 B, wat beteken dat die laaste vier grepe van die laaste monster afgesny is. Die samepersing maak die opname dus 40,64 % kleiner, oftewel 59,35 % van die oorspronklike grootte.

Indien nog spasie benodig word, kan 'n groter SD-kaart gebruik word. Alternatiewelik kan die **ntfs-3g** pakket geïnstalleer word en 'n draagbare hardeskyf by een van die RPI se USB-poorte ingeplug word.

7.13 GPS

Om olifante te kan opspoor, is dit wenslik om die oudiomonsters wat deur twee verskillende apparate opgeneem is, te sinkroniseer. Omdat hierdie opnemers ver van mekaar af in die veld geplaas word, moet hulle battery-aangedrewe en volledig draadloos wees. Oplossings soos die sinkronisasiepoorte wat op videokameras voorkom, is dus buite die kwessie.

GPS-modules bied 'n manier om tyd van satelliete af te verkry, gewoonlik akkuraat binne 100 ns (sien §8.5). Verskeie implementasies van hierdie idee op die RPI bestaan, wat demonstreer dat dit 'n haalbare oplossing is. 'n Besonder gedetailleerde voorbeeld word deur Taylor (2017) gegee. Vir hierdie projek is die **SkyTraq Venus638FLPx** gebruik. Die hardeware word in §8 beskryf.

Die RPI kommunikeer via **UART** (sien §7.3) met die GPS-module. 'n Bykomende **GPIO**-pen moet as **PPS**-pen ingestel word. 'n Puls vind elke sekonde op hierdie pen plaas wanneer die **Venus638FLPx** genoeg satelliete sien om die 3D-ligging te bepaal.

As die **PPS**-pen van die GPS-module hooggetrek word terwyl die module aangeskakel word, betree die module 'n ongewenste programmeringsmodus. Die **GPIO**-pen wat aan **PPS** gekoppel word, moet dus nie 'n optrekweerstand hê nie. **GPIO24** word met behulp van die **DT** as 'n **intreepoort** met 'n aftrekweerstand gestel.

Selfs al word **DTOs** gebruik om die **GPIOs** op te stel, kan die gedrag van die **GPIOs** eers gewaarborg word ná die **Linux**-kern (wat die **DT** lees en **GPIOs** beheer) klaar gelaai is. Om dus heeltemal seker te maak dat die **PPS**-pen nie opgetrek word nie, word 'n diode tussen **PPS** en **GPIO24** geplaas sodat stroom slegs van die GPS-module na die RPI kan vloei (sien

⁵ 'n Variant van **gzip** wat geskryf is om gelyktydig van al die verwerkerkerne gebruik te maak.

Figuur 8.1(a)). Dit maak dit onmoontlik vir die **RPI** om die **PPS**-pen hoog te trek, maar vereis ook dat daar 'n aftrekweerstand by **GPIO24** moet wees sodat die sein na 0 V kan daal wanneer **PPS** ook laag is.

7.14 Tydhou

Die **GPS**-agtergrondproses (**GPSD**, **GPS daemon**) sagteware van Raymond *et al.* (2016) word gebruik om met **GPS**-module te kommunikeer en **PPS**-seine te hanteer.

Daar is twee maniere hoe die **PPS** hanteer kan word (Miller en Raymond, 2016):

Gewone onderbreking 'n Puls maak die **GPSD-draad** wakker, wat dan die bedryfstelsel se klok lees om te bepaal wanneer dit plaasgevind het. Volgens Miller en Raymond (2016) neem dit 'n aansienlike hoeveelheid oorhoofse tyd, wat die akkuraatheid van die **PPS** verminder.

Kern-PPS Die Linux-kern teken die presiese tyd van elke **PPS** aan en maak dit aan ander prosesse beskikbaar om later te lees ('n implementasie van Mogul *et al.*, 2000). Dit bied baie beter akkuraatheid as die gewone metode. Op die **RPI** word die **pps-gpio DTO** gebruik om 'n **GPIO** as 'n kern-**PPS-intree** te registreer. Die Linux-kern maak dan die tydtempels by die virtuele apparaat **/dev/pps0** beskikbaar. Dit is die enigste opsie wat vir hierdie projek oorweeg is.

GPSD word soos volg (met **sudo**) uitgevoer wanneer die **RPI** aanskakel:

```
/usr/local/sbin/gpsd -n /dev/ttyAMA0 /dev/pps0
```

Die volgende opsies word deur Raymond *et al.* (2017a) gedokumenteer:

- **-n** laat kommunikasie met die **GPS** begin, nog voordat enige programme daarvoor vra.
- **/dev/ttyAMA0** is die naam van die **UART** waaraan die **GPS** gekoppel is.
- **/dev/pps0** is die naam van die kern-**PPS**-apparaat waar die tydtempels van die pulse gelees kan word.

Die pulse word dus by **/dev/pps0** gelees om die begin van elke sekonde akkuraat te kan bepaal. Die absolute tyd en ligging word via **/dev/ttyAMA0** ontvang. Die resultaat word in gedeelde geheue (**SHM**, **SHared Memory**) beskikbaar gestel wat met virtuele internetprotokol (**IP**)-adresse in die omvang van 127.127.28.x bereik kan word (Mills, 2014a; Miller en Raymond, 2016).

Die **NTP**-agtergrondproses (**NTPD**, **NTP daemon**) word gebruik om hierdie tydtempels te lees en die bedryfstelsel se horlosie daarvolgens op te dateer.

NTPD kan opgestel word om na verskeie netwerktydprotokol (**NTP**)-bedieners te luister en die korrekte tyd daaruit af te skat. Elke bediener het 'n stratum, of rang, wat aandui hoeveel stappe daar tussen die bediener en die naaste egte verwysingsklok, soos 'n atomiese klok of **GPS**-module, is. Die **GPS** self het dus rang 0, en die **NTP**-bediener waaraan die **GPS** gekoppel is, het rang 1. Bedieners wat hul tyd van 'n eersterangse bediener ontvang, is tweederangs, ensovoorts.

NTPD verwag gewoonlik dat daar meer as twee bedieners is om met mekaar te vergelyk. Die algoritme probeer outomaties bedieners wat die verkeerde tyd gee, uitsluit. As daar slegs een of twee bronne van tyd is, is dit onmoontlik om outomaties te bepaal watter een reg is. Siende dat hierdie projek in 'n omgewing moet funksioneer wat van die internet geïsoleer is, moet **NTPD** aangesê word om altyd die **GPS**-tyd as gesaghebbend te beskou. Volgens Windl *et al.* (2006, §6.1.2.1, §7.1.1) kan dit gedoen word deur die bedryfstelsel se eie horlosie as 'n verwysing met 'n lae rang te gebruik, sodat daar altyd iets is om die **GPS**-tyd mee te vergelyk.

Volgens Windl *et al.* (2006, §5.1.1.4) reageer **NTPD** verskillend op klein, medium en groot verskille tussen die plaaslike tyd (die bedryfstelsel se horlosie) en die verwysingstyd (die bron van akkurate tyd, soos **GPST**):

- Vir verskille kleiner as 128 ms word die plaaslike tyd geleidelik aangepas deur die sekondes te rek of krimp (Windl *et al.*, 2006, §6.3.2.1).
- Vir effens groter verskille word die tyd met 'n enkele sprong gestel.
- Vir baie groot verskille, neem **NTPD** aan dat iets buitengewoons gebeur het en stop. Meer spesifiek, as die plaaslike tyd en verwysingstyd vir langer as ongeveer 20 min meer as 1000 s verskil, sal **NTPD** staak (Windl *et al.*, 2006, §8.2.1.2).

As die **RPI** lank afgeskakel was, is die horlosie se tyd uit met 'n groot sprong, en **NTPD** sal staak. Om dit te voorkom, word **NTPD** met die parameters **-gG** begin:

- **-g** (of **--panicgate**) beveel **NTPD** om nie te staak as die verskil tussen die plaaslike tyd en verwysingstyd aanvanklik groot is nie. Nadat die eerste aanpassing gemaak is, het hierdie parameter nie meer 'n effek nie.
- **-G** (of **--force-step-once**) beveel **NTPD** om die eerste tydaanpassing altyd met 'n enkele sprong te doen, sodat die tyd so gou as moontlik korrek is.

As die **GPS**-module sukkel om die tyd en ligging vas te stel, sal **NTPD** waarskynlik staak. **systemd**⁶ word gestel om **NTPD** na 10 s weer te begin as dit sou stop of **ineenstort**. Sodoende kry die **GPS** tyd om die ligging te bepaal. Daarna behoort **NTPD** korrek te funksioneer en die bedryfstelsel se tyd met die **GPS**-tyd gelyk te hou.

Tabel 7.5 toon die nodige riglyne wat in die lêer by **/etc/ntp.conf** geskryf word, om **NTPD** en **GPST** goed te laat saamwerk.

Twee verdere verbeterings word aan hierdie tydhoudstelsel aangebring:

- Miller en Raymond (2016) beveel aan dat kragbesparingsfunksies afgeskakel word deur **nohz=off** by die kernparameters in **/boot/cmdline.txt** te voeg. Dit verbeter die stabiliteit van die kern-**PPS**. Hierdie parameter word deur Torvalds *et al.* (2017b) gedokumenteer.
- **fake-hwclock** word geïnstalleer. Hierdie program stoor die tyd op die **SD**-kaart wanneer die **RPI** afkakel en stel weer die horlose na daardie tyd wanneer die **RPI** aanskakel, sodat die aanvanklike tyd nader aan die regte tyd is. Dit maak die aanvanklike sprong wat deur **NTPD** gemaak moet word, kleiner. Dit voorkom ook verwarring met lêers wat geskep word voordat die **GPS** die tyd en ligging vasgestel het. Sonder **fake-hwclock** sou die skeppingsdatum van so 'n lêer in die verre verlede wees.

7.15 Skakelaars

Die kragbronbeheerder in §3.5 bevat die skakelaar waarmee die **RPI** aangeskakel kan word (SW1 in Figuur 3.11). Die skakelaar moet egter ingehou word totdat die **RPI GPIO3** kan hoogtrek. **GPIO3** is hiervoor gekies omdat dit 'n fisiese optrekweerstand het, wat beteken dat dit vroeër as enige ander pen 'n hoë waarde sal hê. Sodra die Linux-kern oorneem, word daardie pen as 'n **uittreepen** gestel en hoog gehou. Gevolglik laai kapasitor C1 baie gou, en die relê skakel so te sê onmiddellik aan.

'n **DTO** deur Trønnes (2016) word gebruik om seker te maak dat **GPIO3** laag gemaak word wanneer die **RPI** afkakel. Wanneer dit gebeur, ontlai kapasitor C1 stadig deur die basisstroom

⁶ **systemd** is die program wat agtergrond Dienste soos **NTPD** en **GPST** op 'n Linux-stelsel begin en stop.

Riglyn	Beskrywing
<pre>server 127.127.1.0 noselect fudge 127.127.1.0 stratum 10</pre>	<p>Gebruik die bedryfstelsel se eie horlosie as 'n bron van tyd, sodat daar altyd iets is om die GPS-tyd mee te vergelyk.</p> <ul style="list-style-type: none"> • 127.127.1.0 verwys na die bedryfstelsel-horlosie. • stratum 10 beteken dat die horlosie 'n baie lae rang het en dus nie geredelik vertrou moet word nie. • noselect beveel NTPD om nooit hierdie tydbron as verwysing te kies nie.
<pre>restrict -4 default ignore restrict -6 default ignore restrict -4 127.0.0.1 restrict -6 ::1 server 127.127.28.0 fudge 127.127.28.0 time1 0.08354 fudge 127.127.28.0 refid GPS fudge 127.127.28.0 flag1 1</pre>	<p>Moenie op enige netwerkversoeke reageer nie. Laat slegs prosesse op die RPI self toe om die tyd van NTPD te versoek.</p> <p>Lees die absolute tyd.</p> <ul style="list-style-type: none"> • 127.127.28.0 verwys na die SHM eenheid 0, wat die resultaat van GPSD se kommunikasie met die GPS via UART bevat. • time1 is 'n konstante waarde wat van die tyd afgetrek word om vir vertrags tussens die UART-tyd die PPS-tyd te kompenseer (Mills, 2014b). • refid GPS verbind die naam GPS aan hierdie IP-adres. • flag1 1 beveel die SHM-drywer om groot tydspronge toe te laat (Mills, 2014b). Dit is nodig vir situasies waar die RPI lank afgeskakel was en daar dus 'n groot verskil tussen die plaaslike tyd en die GPS-tyd is.
<pre>server 127.127.28.2 prefer fudge 127.127.28.2 refid PPS</pre>	<p>Lees die presiese PPS-tyd.</p> <ul style="list-style-type: none"> • 127.127.28.2 verwys na die SHM eenheid 2, wat GPSD se interpretasie van die kern-PPS tydtempels bevat. • prefer beveel NTPD om altyd die PPS-tyd bó die UART-tyd te vertrou. • refid PPS verbind die naam PPS aan hierdie IP-adres.

Tabel 7.5: 'n Volledige beskrywing van riglyne in **ntp.conf** wat nodig is vir die samewerking tussen **GPSD** en **NTPD**.

van transistor Q1, totdat die relê 'n paar sekondes later afskakel en sodoende die kragtoevoer na die RPI afsny. Dit keer dat die battery leegloop wanneer die stelsel nie gebruik word nie.

'n Tweede skakelaar word tussen GPIO23 en GND gekoppel. Tydens aansakeling word 'n program uitgevoer wat wag totdat GPIO23 daal en dan die bevel `sudo poweroff` uitvoer om die RPI se bedryfstelsel veilig tot stilstand te bring.

Die batterymonitor in §6.10 maak GPIO13 hoog as die 5 V kragtoevoer laer as ongeveer 4,5 V word. Nog 'n program word tydens aansakeling uitgevoer wat wag totdat hierdie pen styg en dan die bevel `sudo poweroff` uitvoer.

Die bronkode en konfigurasie-lêers vir al die bogenoemde is op DVD E.1 in Bylae E beskikbaar.

7.16 SSH en WiFi

Die RPI word via 'n beveiligde tolk (SSH, Secure SHell) en deur WiFi beheer⁷. Vir hierdie prototipe is daar nie aandag gegee aan sekuriteit nie, en die standaardgebruikersnaam⁸ en -wagwoord⁹ van die Raspbian bedryfstelsel is gehou.

Die pakkette `dbus` en `libpam-systemd` word geïnstalleer om seker te maak dat SSH-verbindings korrek afgesluit word wanneer die RPI 'n afskakelprosedure begin.

Die RPI 3 kan gestel word om outomaties aan beskikbare WiFi-toegangspunte te koppel deur die netwerknaam en -wagwoord by die konfigurasielêer `/etc/wpa_supplicant/wpa_supplicant.conf` te voeg:

```
network={
    ssid="OlifantDraadloos"
    psk=...
    priority=10
}
```

Om die RPI maklik op die netwerk te kan vind sonder om die IP-adres in te tik, moet Windows Internet Name Service (WINS) by die opsies in `/etc/nsswitch.conf` gevoeg word:

```
...
hosts: files wins mdns4_minimal [NOTFOUND=return] dns
...
```

7.17 Outomatiese opstelling van 'n nuwe RPI

Om 'n Linux-stelsel vir 'n ingebodde hardware-omgewing op te stel is meer ingewikkeld as die programmering van 'n mikrobeheerder soos die PIC32. Daarom is 'n installeerprogram geskryf wat alle sagtewarevoorvereistes vir hierdie projek op 'n skoon Raspbian-omgewing kan installeer. Die skripte hiervoor is op DVD E.1 in Bylae E beskikbaar.

Die `install.sh` program berei die stelsel soos volg voor:

1. Gradeer die bedryfstelsel en alle pakkette met `apt-get` op.
2. Installeer die pakkette in Tabel 7.6 met behulp van `apt-get`.
3. Verwyder die ou weergawe van NTP wat saam met die bedryfstelsel kom (sien §7.14). Stop ook `systemd-timesyncd.service` indien dit bestaan, omdat dit sal bots met NTPD.
4. Laai bronkode vir die pakkette in Tabel 7.7 af en bou dit.

⁷As die projek op 'n RPI 2 geïmplementeer word, kan SSH oor gewone ethernet gebruik word.

⁸pi

⁹raspberry

Pakket	Doel	§
scons	GPSPD-voorvereiste	7.13
libncurses5-dev	GPSPD-voorvereiste	7.13
python-dev	GPSPD-voorvereiste	7.13
pps-tools	GPSPD-voorvereiste	7.13
fake-hwclock	Tydhou	7.14
ssh	Toegang deur netwerk	7.16
libpam-systemd	Sluit SSH-verbindings as RPI afskakel	7.16
dbus	Sluit SSH-verbindings as RPI afskakel	7.16
ntfs-3g	Gebruik NTFS-media via USB	7.12
git	Laaï bronkode af	–
pv	Voorvereiste vir die opnemerprogram	7.12 & 7.7
pigz	Voorvereiste vir die opnemerprogram	7.12 & 7.7

Tabel 7.6: Pakkette wat met `apt-get` geïnstalleer word.

Pakket	Beskikbaar by	Doel	§
BCM2835 1.50	McCauley (2016)	SPI	7.4
WiringPi	Henderson (2017)	Opstelling van poorte	7.1
NTP 4.2.8p10	Network Time Foundation (2017)	Tydhou	7.14
GPSPD 3.16	Raymond <i>et al.</i> (2016)	GPS en tydhou	7.13, 7.14
Buffer-drywer	DVD E.1	Buffer	7.4, 7.5, 7.6, 7.7

Tabel 7.7: Pakkette wat uit bronkode gekompileer word.

5. Skakel die **ruilgeheue** af.
6. Installeer die lêers in Tabel 7.8, of dateer dit op indien dit reeds bestaan:
7. Kompileer en installeer 'n **DTO** wat seker maak dat **GPIOs** 23 en 24 (penne 16 en 18 – sien §7.1) so gou as moontlik deur die Linux-kern as **intreepoorte** gestel word – **GPIO** 23 met 'n optrekweerstand en **GPIO** 24 met 'n aftrekweerstand (Byker, 2017).
8. Aktiveer die bogenoemde dienste met behulp van `systemctl`.

As `install.sh` suksesvol uitgevoer is, behoort die **RPI** gereed te wees om aan die res van die hardeware verbind te word. Bylae D toon stapsgewys hoe die bogenoemde installeerprogram gebruik kan word om 'n nuwe Raspbian-stelsel op te stel vir gebruik met hierdie projek.

Daar is ook volledige **afbeeldings** van die **SD**-kaart van elke opnemer wat in hierdie projek gebruik is op DVD E.5 in Bylae E. Bylae C wys hoe hierdie **afbeeldings** gemaak is, en hoe om dit na 'n **SD**-kaart te herstel vir verdere gebruik.

7.18 Toekomstige verbeterings

Die bostaande ontwerp is funksioneel, maar laat baie ruimte vir verbetering. Die volgende voorstelle word gemaak vir toekomstige werk aan hierdie deel van die projek:

- Skryf sagteware wat die toestand van die sagteware **intyds** rapporteer. Dit sou die volgende funksies kan insluit:
 - Toon hoe vol alle buffers is.

Lêer	Doel	§
/boot/...		
cmdline.txt	Linux-kernparameters	7.2
config.txt	DT	7.2
/etc/...		
nsswitch.conf	Vind netwerkname met WINS	7.16
ntp.conf	NTPD-instellings	7.14
ssh/sshd_config	SSH-instellings	7.16
systemd/system/...		
gpsd.service	Begin GPSD outomaties	7.14
log_location.service	Begin log_location.sh outomaties	–
monitor_health.service	Begin monitor_health.sh outomaties	7.15
ntpd.service	Begin NTPD outomaties	7.14
power_off_button.service	Begin power_off_button.sh outomaties	7.15
record.service	Begin klankopnames outomaties	–
wpa_supplicant/...		
wpa_supplicant.conf	Stel WiFi in	7.16
/home/pi/...		
log_location.sh	Skryf die huidige ligging periodies na die SD-kaart.	–
monitor_health.sh	Skakel veilig af as temperatuur hoog of battery swak is.	7.15
power_off_button.sh	Skakel veilig af as 'n knoppie gedruk word	7.15
recording_restart.sh	Stop en hervat klankopname	–
recording_start.sh	Roep app_stream_all_channels en stuur data na SD-kaart	7.7
recording_status.sh	Bekyk die toestand van die opname-program	–
recording_stop.sh	Stop die klankopname	–

Tabel 7.8: Konfigurasie-lêers wat geïnstaleer of opgedateer word. Al hierdie lêers is by die bronkode op DVD E.1 in Bylae E beskikbaar.

- Waarsku as die inkomende klanksein te hard is en afgesny word.
- Laat die gebruiker toe om opnames te stop en te begin.
- Soek outomaties na USB-apparate om opnames op te stoor, en laat die gebruiker kies waar om opnames te stoor.
- Laat die gebruiker toe om opnames via 'n webkoppelvlak af te laai.
- Verbeter die bronkode om minder interafhanklik en meer modulêr te wees.
- Maak die oudiodata op 'n nie-blokkerende stroom (soos 'n UDP-poort) beskikbaar, sodat daar intyds na die klankseine geluister kan word. So 'n stroom waarborg nie dat alle monsters behoue bly nie, maar is baie nuttig vir ontfoutingsdoeleindes, soos om te inspekteer of al die mikrofone nog werk.
- Indien dit nodig sou wees om meer verwerking op die RPI te doen en verwerkingstyd word skaars, kan die volgende optimerings oorweeg word:

- Gebruik **DMA** vir **SPI** aan die **RPI**-kant.
- Laat die interpretasie van monsters in 'n ander **draad** gebeur as die **SPI**-transaksies. Alle **drade** moet dan 'n laer prioriteit hê as die een wat die **SPI**-transaksies behartig. Dit verhoog wel die kompleksiteit van die sagteware, omdat daar by elke stap 'n buffer benodig word waardeur data van een **draad** aan 'n volgende oorhandig word.
- Maak dit moontlik om die sagteware in die veld op te dateer, sonder om die apparaat af te skakel.
- Verlaag die frekwensie van die **RPI** se verwerker om batterykrag te spaar.

Hoofstuk 8

GPS-module

Om klankbronopsoring te doen, moet die opnames van verskeie onafhanklike eenhede gesinkroniseer word. 'n **GPS**-module bied akkurate inligting omtrent die tyd en ligging van die toestel.

GPS-modules het gewoonlik twee koppelvlakke:

- **PPS** is 'n pulstrein waarvan die **opgaande flank** nouliks met die sekondeverandering op die **GPS**-satelliete ooreenstem. Die **dienssiklus** van die pulstrein is nie gestandaardiseer nie, maar is gewoonlik minder as 50 %. Die **PPS-uitree** bied die mees akkurate tyd, maar dui slegs die begin van die sekonde aan.
- Meer gedetailleerde inligting, soos die datum, absolute tyd en die posisie, word via 'n serie-poort soos **UART** gekommunikeer.

Die **GPSD**-sagteware op die **RPI** lees beide bronne van tyd-inligting en stel die **Linux**-horlosie daarvolgens. Die **RPI** hou dan die **PIC32MX** se horlosie dop en stel dit reg indien die absolute tyd meer as 3 s afwyk. Dit is slegs nodig wanneer die mikrobeheerder aangeskakel of herstel word (sien Figure 7.5, 7.9 en 7.10) en as die **GPS**-sein weggeval het (byvoorbeeld in slegte weer, wanneer die **GPS**-satelliete nie meer sigbaar is nie). Die presiese tyd op die **PIC32MX** word onder normale omstandighede deur die **PPS**-pen beheer: Die mikrobeheerder tel een sekonde aan vir elke **opgaande flank** van die **PPS**-sein (sien Tabel 6.6). Die sub-sekonde akkuraatheid word deur die ossilator van die versyferaar verskaf: 'n Ranggetal word aan elke monster toegeken. Elke keer as 'n **PPS** plaasvind, begin hierdie ranggetal weer by 0. Die tydstempel en ranggetal word saam met elke monster gestoor. Dit is dus moontlik om lang opnames volledig te sinkroniseer, selfs al was daar nie 'n betroubare **GPS**-verbinding vir die volle duur van die opname nie.

Hierdie strategie werk op die aanname dat die frekwensie van die versyferaar se kristal konstant bly oor 'n kort tyd. Omdat die tydstempel en ranggetal saam met elke monster gestoor word, kan die langtermyn-gedrag van daardie kristal ook geassesseer word om die aanname te toets. Indien die versyferaar se klok byvoorbeeld stadiger as normaal sou loop, sal die ranggetalle minder as 4800 (die **monstertempo**) wees voordat dit weer by 0 begin. Indien die **GPS**-module akkurate ligging en tyd verskaf terwyl dit gebeur, sal dit selfs moontlik wees om 'n fout in die **monstertempo** ná die tyd te korreger.

Hierdie hoofstuk bespreek die keuse van 'n **GPS**-module vir hierdie projek en die verbindings met die ander stelselkomponente soos die **RPI** en **PIC32MX**. Die akkuraatheid van die **GPS**-module se tydseine en die geskiktheid van **GPS** as 'n sinkronisasiemetode word ook bespreek. Voorstelle vir verbetering word aan die einde van die hoofstuk gemaak.

8.1 Keuse van module

Dit is buite die omvang van hierdie projek om beskikbare **GPS**-modules te vergelyk om die beste moontlike **GPS** uit te soek. Die module moet wel aan die volgende vereistes voldoen:

- Dit moet 'n **PPS-uittrepen** hê. Hierdie pulstrein word deur die PIC32MX gebruik om akkurate tydstempels aan elke monster toe te ken.
- Die akkuraatheid van die **opgaande flanke** van die **PPS**-sein moet heelwat korter as een monsterperiode ($1/4800\text{ Hz} \approx 208,3\text{ }\mu\text{s}$) wees. Daar is baie kommersieel beskikbare **GPS**-modules wat 'n akkuraatheid van $< 100\text{ ns}$ bied.
- Dit moet boodskappe volgens **National Marine Electronics Association: spesifikasie 0183 (NMEA0183)** met tyd en ligging via **UART** aan die **RPI** kan stuur om deur **GPSD** geïnterpreteer te word. **GPSD** gebruik die **NMEA0183**-boodskappe saam met die **PPS**-sein om die tyd op die **RPI** akkuraat te stel.
- Die **UART** moet met 3,3 V spanningsvlakke werk om met die **RPI** versoenbaar te wees.
- Daar moet 'n **subminiatur A (SMA)** verbinding wees waaraan 'n **GPS**-antenna gekoppel kan word.
- Dit moet van die beskikbare 3,3 V of 5 V bronne gebruik kan maak.

Sparkfun vervaardig 'n **GPS**-ontwikkelingsbordjie wat aan hierdie vereistes voldoen¹. Dit maak van die SkyTraq Venus638FLPx **GPS**-module gebruik. Die bordjie en module bied ook die volgende nuttige funksies:

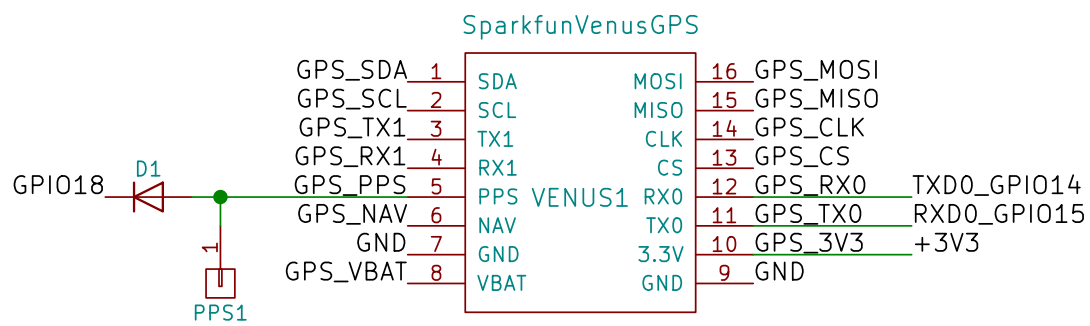
- 'n Superkapasitor kan op die bord bevestig word wat die **GPS**-module se geheue aangeskakel hou. Sodoende bly die **GPS**-almanak vir 'n paar uur in geheue behoue nadat die stelsel afgeskakel is. As die module dan weer aangeskakel word, kan die tyd en ligging binne enkele sekondes bepaal word, omdat dit nie nodig is om na satelliete te soek nie. Volgens die datablad (SkyTraq Technology, Inc., 2011c) neem die **GPS**-module 29 s om die posisie vas te stel na 'n 'koue begin' (met 'n leë superkapasitor) en ongeveer 1 s na 'n 'warm begin' (met 'n gelaaide superkapasitor). Dit is getoets deur die **GPS** aan die **RPI** te koppel en 'n program te loop wat die **bedryfstyd** na 'n lêer skryf sodra die **GPS** module 'n akkurate tyd en ligging vasgestel het. Die aparate is in 'n oop veld neergesit, en die wagtye vir 'n koue en 'n warm begin was onderskeidelik 36 s en 7 s.
- Daar is 'n hoë-drywing en lae-drywing modus. Daar kan dus gekies word tussen beter werksverrigting, of laer stroomverbruik (en dus langer batteryleeftyd).
- Die module bevat 'n **rugsteun-selflaaiprogram**, vir in geval die eerste een faal.
- Dit sal werk met 'n aktiewe of 'n passiewe antenna.

8.2 Verbindings

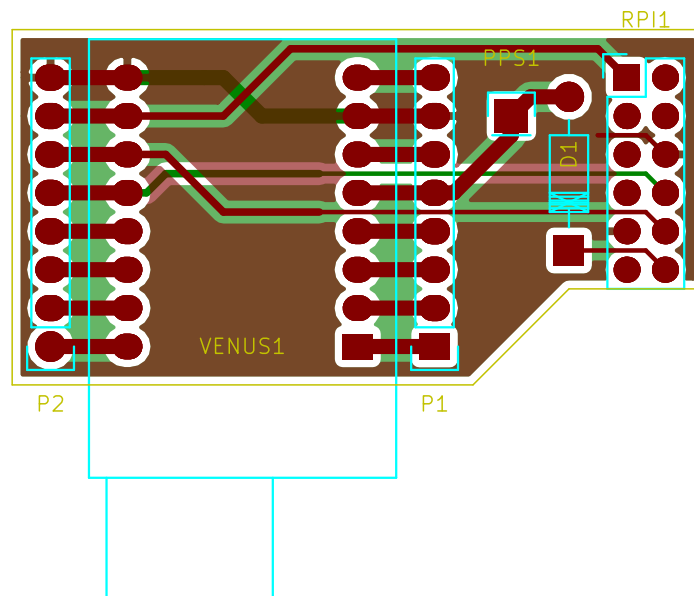
Om hierdie ontwikkelingsbordjie aan die **RPI** te koppel, is die bordjie in Figuur 8.1 ontwerp en vervaardig. Om die Venus638FLPx te herprogrammeer, kan die **PPS**-pen hooggetrek word terwyl die module aanskakel. Diode D1 verhoed dat dit gebeur, maar laat nog steeds die **RPI** se **intrepen** toe om die pulse te lees. Die diode veroorsaak wel 'n spanningsval, sodat 3,3 V by die **PPS**-pen slegs 2,6 V by **GPIO18** veroorsaak, maar dit is hoog genoeg om deur die **RPI** as 'aan' beskou te word.

Kommunikasie tussen die **GPS**-module en die **RPI** word in §7.3 en §7.13 beskryf. Die gebruik van die **PPS**-pen deur die PIC32 word in §6.8 bespreek. Figuur 8.2 toon hoe die **GPS**-module gebruik word om die tyd van beide die **RPI** en die PIC32 te stel, sodat elke monster 'n akkurate tydstempel kan ontvang.

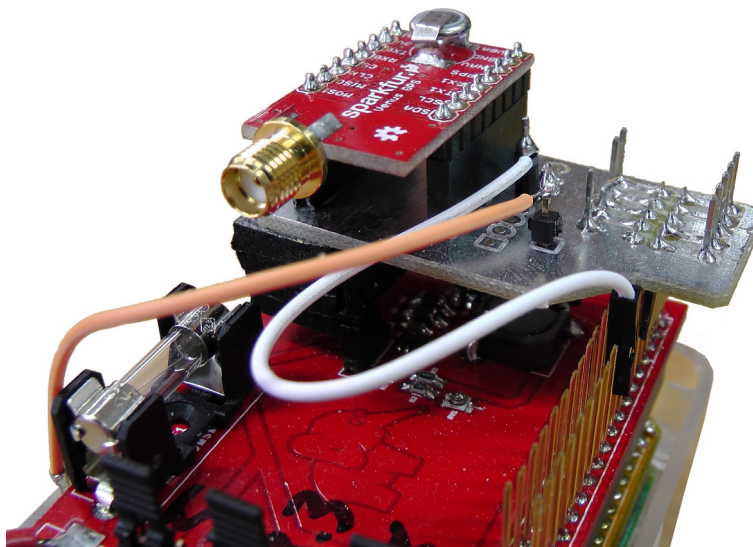
¹ Besikbaar by <https://www.sparkfun.com/products/11058>.



(a) Stroombaandiagram.

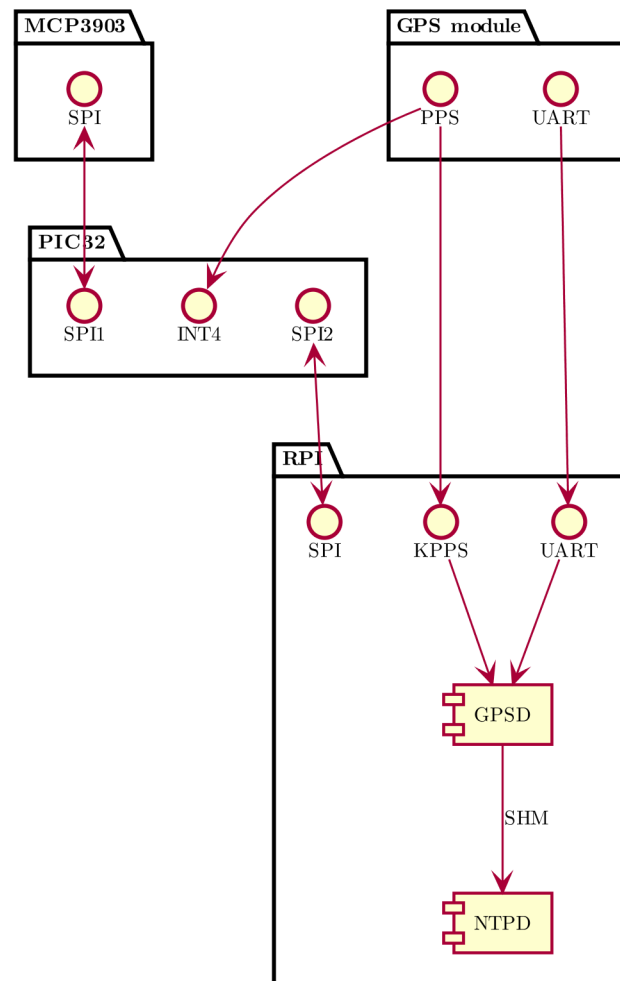


(b) Borduitleg.

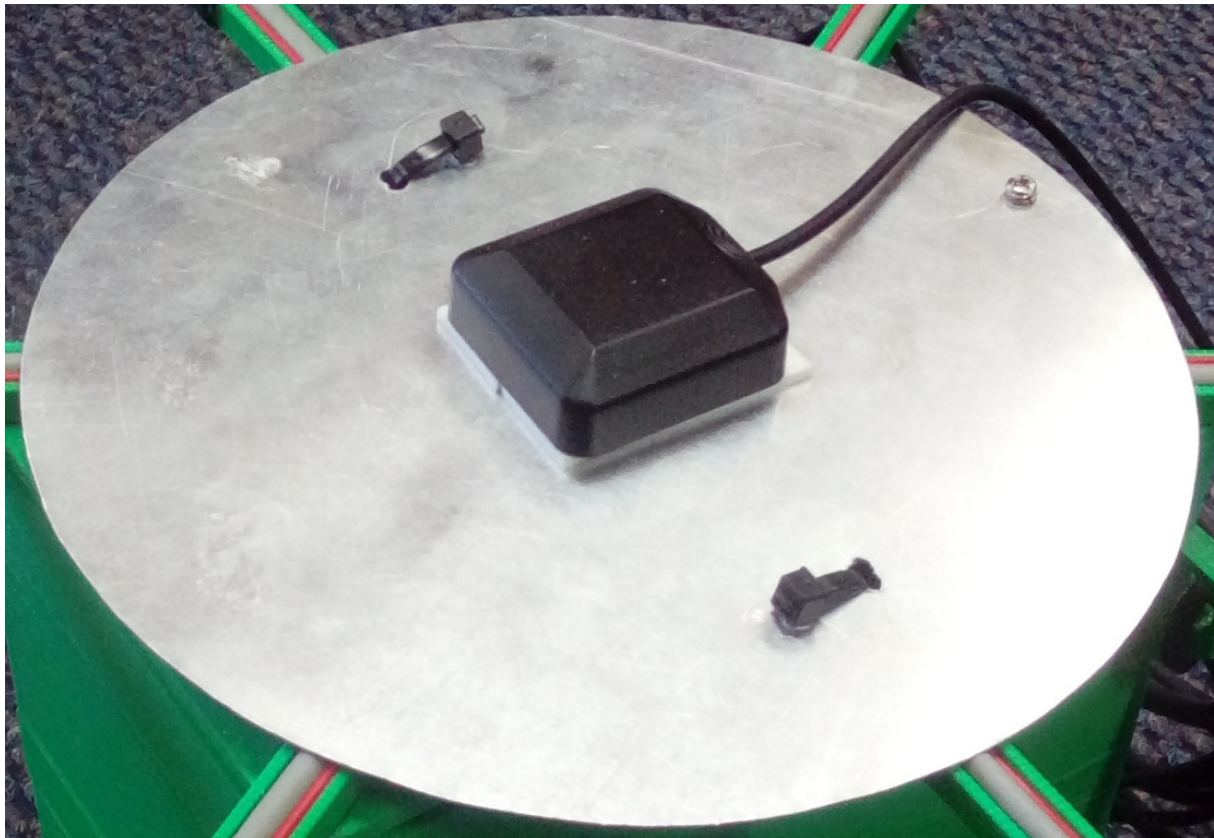


(c) GPS-module bo-op die RPI ingeprop. Die bordjie in die foto is 'n vorige iterasie en verskil dus effens van die uitleg in (b).

Figuur 8.1: 'n Klein stroombaan is ontwerp om die GPS-ontwikkelingsbordjie maklik op die RPI te kan inprop. Diode D1 keer dat GPIO18 die GPS-module se PPS-pen hoogtrek.



Figuur 8.2: Interaksie tussen hardewarekomponente met betrekking tot sinkronisasie. Die **GPS**-module stuur inligting omtrent die huidige tyd en ligging via **UART** na die **RPI**, waar dit deur **GPSD** gelees word. Die **PPS**-sein word beide deur die **RPI** en die **PIC32** gelees. Op die **RPI** reageer die **Linux**-kern op die pulse en maak die tydstempels van die pulse aan **GPSD** beskikbaar. **NTPD** ontvang die tyd-inligting via 'n **SHM** van **GPSD** en maak seker dat die bedryfstelsel se tyd presies in lyn loop met die **GPS**-tyd. Die tyd op die **PIC32** word met tussenposes getoets en via **SPI** opgedateer as dit meer as 3 s afwyk. Die sekondeteller op die **PIC32** word aangeskuiw met elke puls wat op **INT4** verskyn. 'n Akkurate tydstempel word dan by elke monster gevoeg sodra dit van die versyferaar ontvang word.



Figuur 8.3: Die GPS-antenna word op 'n aluminiumplaatjie gemonteer. Die aluminiumplaatjie dien as 'n grondplaat om **multipadversteurings** te voorkom en sodoende die akkuraatheid te verhoog. Die plaat word elektries aan die **GND** van die res van die stelsel gekoppel. Omdat aluminium nie vir gewone soldeerwerk vatbaar is nie, word die draad aan die plaat vasgebout.

8.3 Antenna

Die GPS-modulevervaardiger **u-blox** (2009), asook die outeur van **GPSD**, **Raymond et al.** (2017b), beveel beide aan dat 'n eksterne GPS-antenna gebruik word wat op 'n groot grondvlak, naby die fisiese grond en so ver as moontlik van enige weerkaatsende oppervlaktes af gemonteer word. Dit maak die **foutradius** (**CEP**, **Circular Error Probability**) kleiner en verlaag veral die **gebibber** van die **PPS**-sein, wat voordelig is vir sinkronisasie tussen die klankopnemers.

'n Voor die hand liggende bron van weerkaatsings is byvoorbeeld 'n muur langs die GPS-antenna. Dit is beter om die antenna in 'n oop veld neer te sit.

Om die apparaat teen olifante te beskerm, sal dit waarskynlik hoog bo die grond geplaas moet word. Dit veroorsaak egter dat die sein van die aarde af weerkaats en 'n **multipadversteuring** veroorsaak (**Raymond et al.**, 2017b). Matige weerkaatsings verswak die sein wat van die GPS-satelliet ontvang word, sodat die ligging moeilik vasgestel kan word. Indien die weerkaatsings baie groot is, kan dit vir die direkte sein aangesien word, en 'n verkeerde oplossing vir die ligging en tyd sal vasgestel word. Hierdie probleem kan opgelos word deur 'n groot grondvlak onder die antenna te plaas.

8.4 Konfigurasie

Die **SkyTraq Venus638FLPx** het verskeie opsies wat gestel kan word (**SkyTraq Technology, Inc.**, 2011a). **SkyTraq Technology, Inc.** (2011b) bied 'n program² wat hul GPS-modules se opstelling

² Die program is vir **Windows** geskryf, maar werk deur **Wine**, mits dit opgestel is om toegang tot **Linux** se **UART** te verkry.

Naam	Lengte	UTC-datum	UTC-tyd	Ligging	Geldigheid
GGA	72 B	x	✓	✓	✓
GLL	51 B	x	✓	✓	✓
GSA	61 B	x	x	x	✓
GSV	44–83 B	x	x	x	x
RMC	72 B	✓	✓	✓	✓
VTG	40 B	x	x	x	✓

Tabel 8.1: Beskikbare **NMEA0183**-boodskappe wat via **UART** gestuur kan word. Slegs boodskap **RMC** bevat al die nodige inligting wat **GPSD** nodig het om die datum, tyd en ligging te bepaal. Die boodskap bevat ook 'n vlaggie wat die geldigheid van die inligting aandui, sodat **GPSD** die inligting sal verwerp in die geval waar daar nog nie genoeg satelliete opgespoor is om die tyd en ligging vas te stel nie.

via **UART** kan verander. 'n **USB-na-UART-omsetter**³ kan gebruik word om die module aan 'n gewone rekenaar te koppel.

Intussen het **Pbal (2017)** 'n **Arduino**-biblioteek hiervoor geskryf, wat met effense aanpassing ook op die **RPI** sal werk. 'n Toekomstige verbetering is dus om die opstelling van die **GPS**-module ook deel van die installeringsprogram in §7.17 te maak.

Verskeie **NMEA0183**-boodskappe kan periodies via **UART** gestuur word. Hulle word in Tabel 8.1 opgesom. Vir elke boodskap kan 'n periode in sekondes bepaal word. Al die boodskappe kan egter nie gelyktydig geaktiveer word nie, want die **UART**-transaksies moet klaar wees voordat die volgende **PPS** plaasvind. As die boodskappe te lank neem, sal **GPSD** die **PPS** met die verkeerde tyd assosieer, en die tyd op die **RPI** sal uit wees met 'n heelgetal aantal sekondes. Die **RMC**-boodskap bied al die nodige inligting (**SkyTraq Technology, Inc., 2010**) en word gestel om elke sekonde gestuur te word. Die ander boodskappe word afgeskakel.

Die aarde is nie 'n presiese sfeer nie, en daarom is dit nodig om aanpassings aan die **GPS** se posisie-skatting te maak vir sekere dele van die aarde. 'n Geodetiese datum is 'n koördinaatstelsel wat die aarde as 'n ellipsoïed met 'n translasië modelleer. Volgens **Zaca (2015)** is die amptelike model vir Suid-Afrika sedert 1999 die 'Hartebeesthoek 1994' datum. Die **Venus638FLPx** bevat egter 'n ouer databasis wat die verouderde 'Cape' datum vir Suid-Afrika gebruik. Die gebruik van die korrekte datum verbeter oor die algemeen die vasgestelde ligging (**Raymond et al., 2017b**). Dit is onseker tot hoe 'n mate die verskil in vasgestelde ligging ook die vasgestelde tyd sal beïnvloed.

Die gerapporteerde ligging van 'n stilstaande **GPS**-module sal altyd effens rondskuif as gevolg van faktore soos ruis. Die **Venus638FLPx** bied 'n 'vaspen' funksie, wat die berekende posisie sal vries as die module langer as 'n sekere tyd binne 'n sekere radius bly en stadiger as 'n sekere spoed beweeg (**SkyTraq Technology, Inc., 2013**). Hierdie funksie word aangeskakel en die verstekparameters word behou:

$$\text{PSPEED} = 2 \text{ km} \cdot \text{h}^{-1}$$

$$\text{PTH} = 10 \text{ s}$$

$$\text{UPSPEED} = 8 \text{ km} \cdot \text{h}^{-1}$$

$$\text{UPTH} = 45 \text{ s}$$

$$\text{UPDISTANCE} = 500 \text{ m}$$

As die module langer as 10 s stadiger as $2 \text{ km} \cdot \text{h}^{-1}$ beweeg, sal dit die vaspen-modus betree. Dit moet dan verder as 500 m beweeg, of vir meer as 45 s vinniger as $8 \text{ km} \cdot \text{h}^{-1}$ beweeg om weer na die normale modus terug te keer.

Daar kan ook gekies word tussen 'n 'voetgangermodus' en 'n 'voertuigmodus'. Die voetgangermodus word gekies omdat dit waarskynlik beter sal werk vir 'n stilstaande module. Die datablad gee geen verduideliking van hierdie modusse nie.

³ Die omsetter moet met 3,3 V-spanningsvlakke werk. **UART**-apparate wat die **RS232**-protokol volg, werk met spannings tot en met $\pm 15 \text{ V}$, wat skadelik is vir die **GPS**-module.

Tydbron	Akkuraatheid
GPS-atoomhorlosie	± 50 ns
Venus638FLPx PPS	± 60 ns
Linux-kern-PPS	± 1 μ s
Onderbrekingsgedrewe PPS	± 5 μ s
USB 2 GPS-module	± 100 μ s
USB 1.1 GPS-module	± 1 ms
NTP-bediener	± 30 ms

Tabel 8.2: Tipiese akkuraatheid van verskillende tydbronne volgens Miller en Raymond (2016). Die spesifikasie vir die akkuraatheid van die Venus638FLPx se PPS is ook ingesluit.

Die laaste opsie wat gestel is, is om die NMEA0183-boodskappe met die PPS te sinkroniseer, sodat die UART-transaksies so gou as moontlik na die puls sal begin.

8.5 Akkuraatheid

GPS is meer akkuraat vir snelheid as posisie, omdat daar van die Doppler-effek gebruik gemaak word om die snelheid te bepaal (Zhang *et al.*, 2006). Volgens die datablad (SkyTraq Technology, Inc., 2011c) het die Venus638FLPx 'n CEP⁴ van 2,5 m en is die beweging akkuraat tot op $0,1 \text{ m} \cdot \text{s}^{-1}$. Die PPS is na bewering akkuraat tot op 60 ns, oftewel een 3472ste van die monsterperiode. Miller en Raymond (2016) gee 'n opsomming van die akkuraatheid van verskillende tydbronne, wat in Tabel 8.2 weergegee word.

Die akkuraatheid van die tyd wat deur die GPS-module ontvang word, het 'n direkte impak op die akkuraatheid van die vasgestelde ligging en andersom. Voor 2 Mei 2000 is daar doelbewuste ruis by alle GPS-data gevoeg as 'n militêre voorsorgmaatreël van die lugmag van die VSA. Toe die regering van die VSA onder President Bill Clinton in die jaar 2000 besluit het om hierdie ruis af te skakel, het die CEP van 'n tipiese stilstaande GPS-ontvanger van 95 % binne 45 m na 95 % binne 6,3 m verbeter (NCO, 2016).

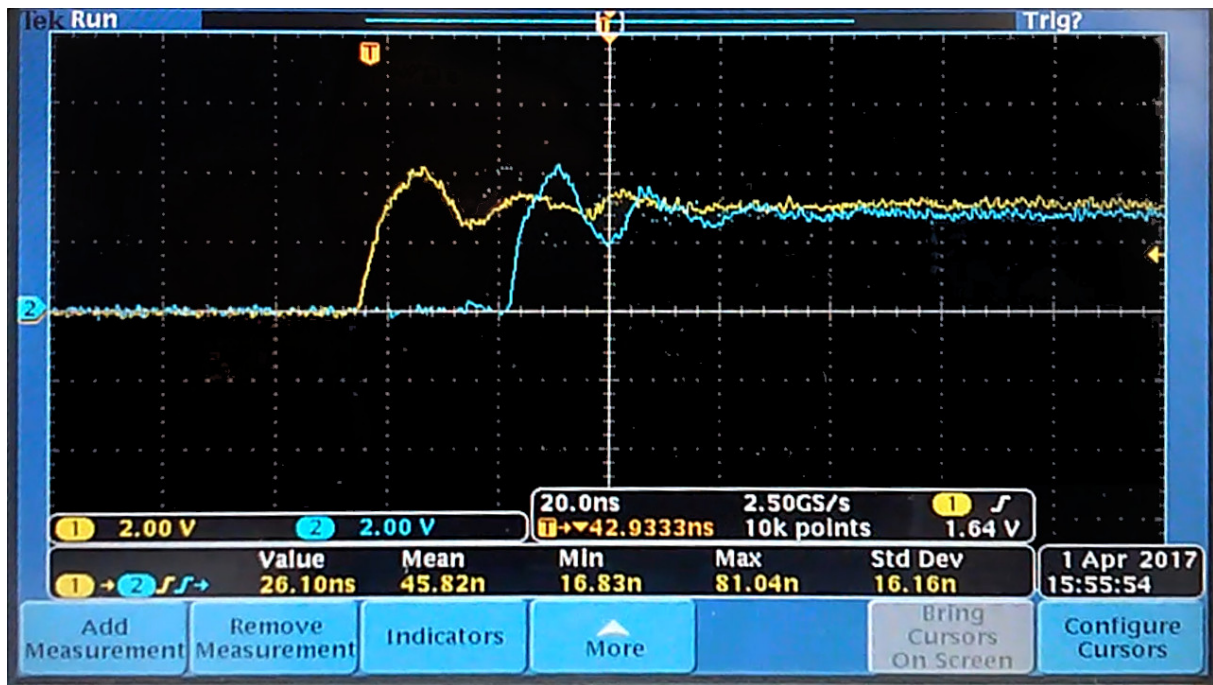
Twee verskillende GPS-modules, a) Navilock GNS602 (wat nie meer vervaardig word nie) en b) SkyTraq Venus638FLPx (wat vir hierdie projek gekies is), is langs mekaar naby 'n venster geplaas. Die PPS-pen van elkeen is aan 'n ossiloskoop gekoppel om die vertraging tussen die opgaande flanke van die onderskeie PPS-seine te meet. Buiten die gemene GND as gevolg van die ossiloskoop se probes, is die stroombane geensins aan mekaar gekoppel nie. Figuur 8.4 toon die skerm van die ossiloskoop tydens een van die metings. Na 'n paar uur het die ossiloskoop die volgende statistieke gerapporteer:

$$\begin{aligned}
 \text{minimum} &= -118,4 \text{ ns} \\
 \text{maksimum} &= 54,15 \text{ ns} \\
 \text{gemiddeld} &= -18,8 \text{ ns} \\
 \text{standaardafwyking} &= 31,3 \text{ ns}
 \end{aligned}$$

Die relatiewe akkuraatheid van die onafhanklike GPS-modules is dus na wense. Soortgelyke akkuraatheid word verwag met identiese modules.

Volgens Miller en Raymond (2016) gebruik GPS 'n 10-bis teller vir die aantal weke sedert 6 Januarie 1980 en 'n aparte teller vir die aantal sekondes sedert die begin van die huidige week. Dit beteken dat die week-teller elke 1024ste week weer by nul begin. Dit het in Augustus 1999

⁴ Daar is verskillende maniere om CEP te definieer en die datablad van die Venus638FLPx vermeld nie hoe dit gedefinieer is nie. Dit is redelik om aan te neem dat hier van 'n 50 % foutsirkel gepraat word. Dit beteken dat 50 % van die liggingsberamings binne 2,5 m van die ware ligging is.



Figuur 8.4: 'n Tektronix DPO3034 ossiloskoop is gebruik om statistieke te versamel omtrent die vertraging tussen die **PPS**-seine van twee verskillende en onafhanklike **GPS**-modules.

gebeur en sal weer in April 2019 gebeur. Die **GPS**-module sal dus na 2019 met 'n nuwer model vervang moet word.

Skrikkelsekondes is nog 'n bron van onakkuraatheid. **Gekoördineerde universele tyd (UTC)** is per definisie 'n kompromie tussen die aarde se rotasie om sy eie as en die tyd wat deur atoomklokke gedefinieer word. Om **UTC** binne 1 s van die rotasietyd te hou, word daar aan die einde van sommige maande 'n skrikkelsekonde bygevoeg. Wanneer 'n **GPS**-module sopas aangeskakel is, is dit nog nie bewus van die skrikkelsekonde-aanpassings nie. Hierdie aanpassings word elke 12,5 min deur die **GPS**-sattelite uitgesaai (Miller en Raymond, 2016).

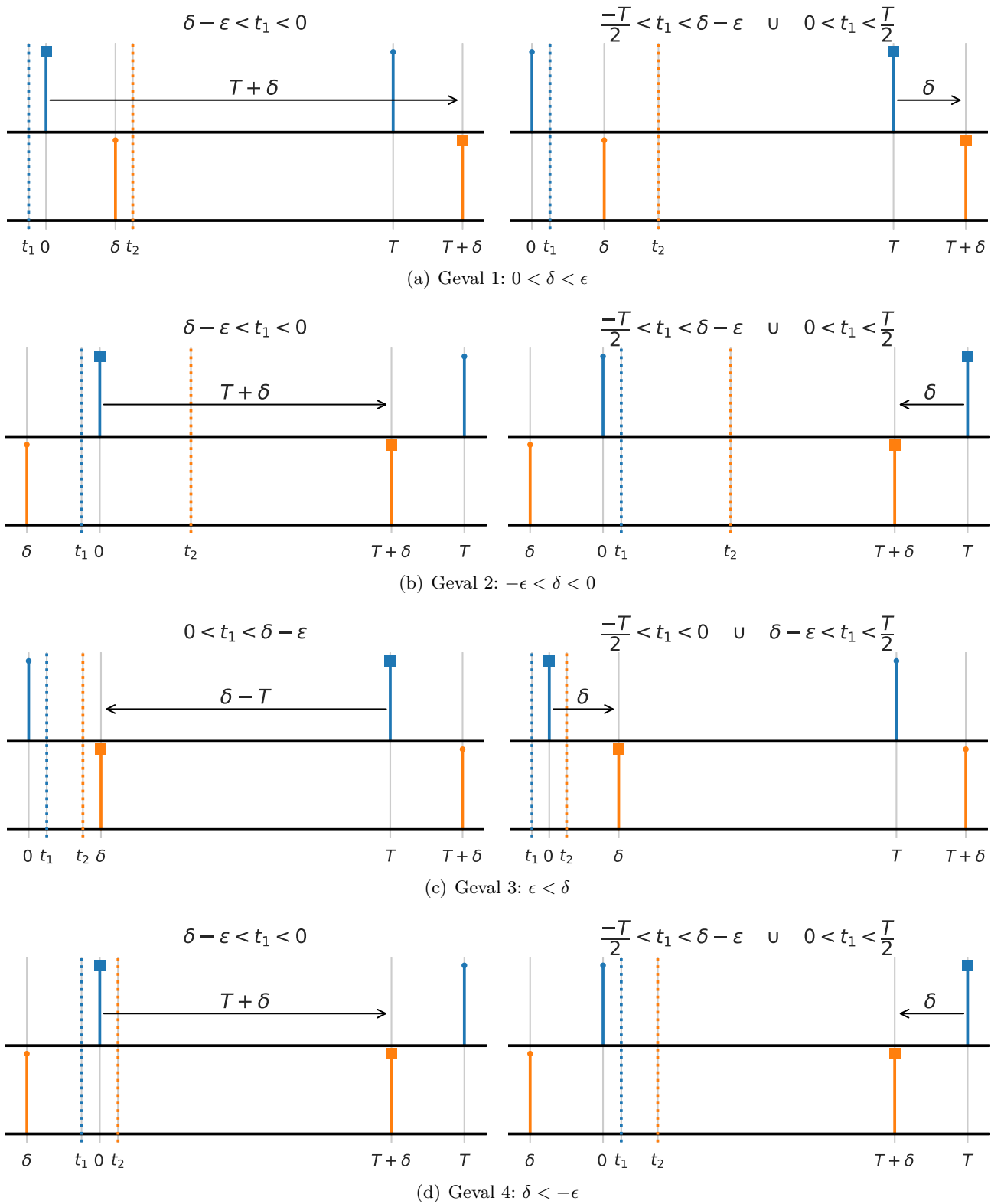
8.6 Wiskundige ontleding van sinkronisasiefout

Selfs wanneer die **PPS**-seine baie min van mekaar afwyk, bestaan die moontlikheid steeds dat monsterring tussen twee pulse wat dieselfde sekonde aandui, plaasvind. Die kristalle van die onafhanklike versyferaars is ook nie sinkroon nie.

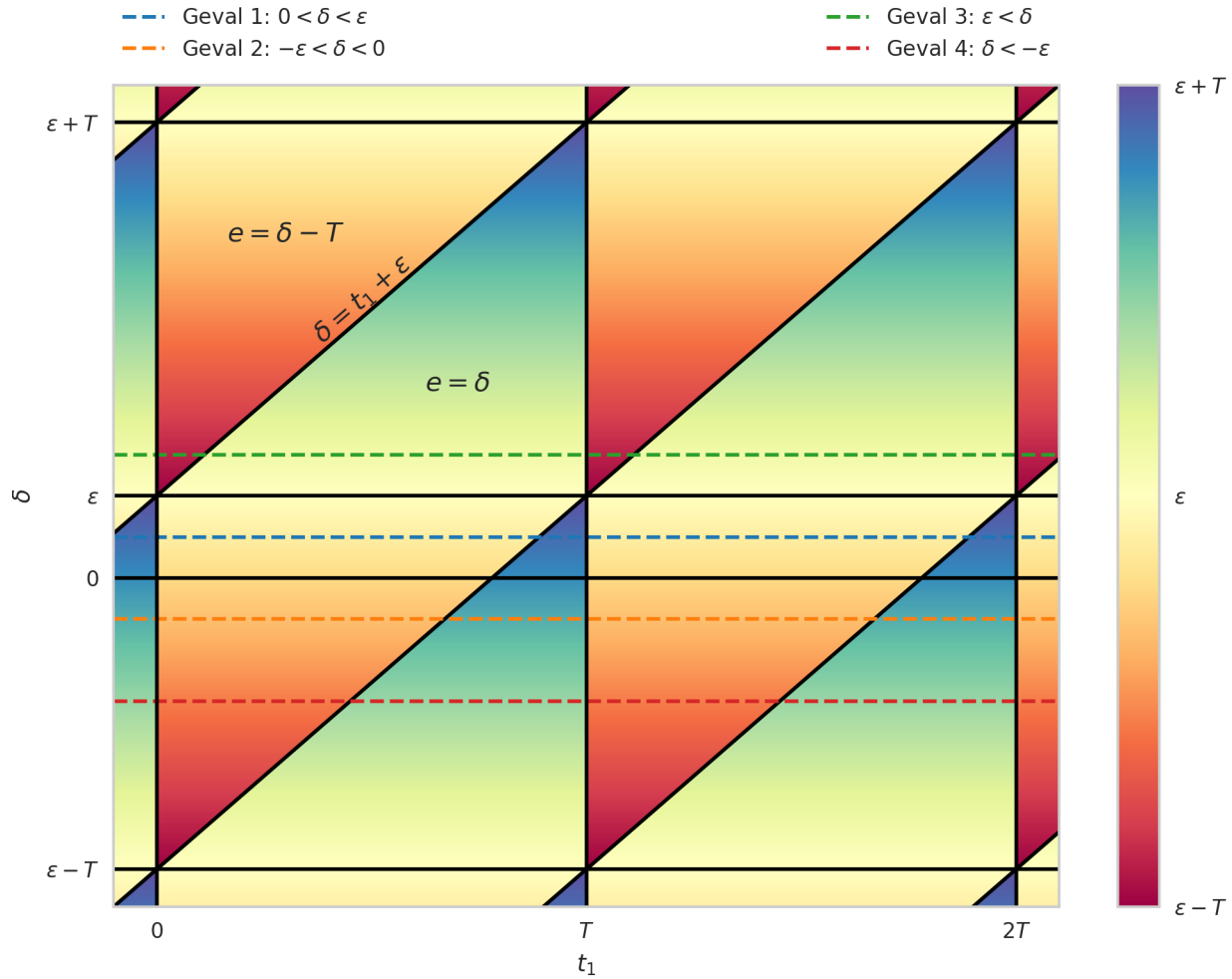
Beskou twee onafhanklike versyferaars met monsterperiode T . Definieer die oorsprong van die tyd-as as enige tydstep van versyfering by versyferaar 1. Laat δ die tydverskil tussen die tydstep van versyfering by versyferaar 1 en die tydstep van versyfering by versyferaar 2 verteenwoordig. Laat t_1 en t_2 die tydsteppe van die opgaande **PPS**-flanke by die onderskeie versyferaars wees. Definieer die tydverskil $\epsilon = t_2 - t_1$ en laat $\epsilon > 0$. Dan kan die sinkronisasiefout e in vier gevalle opgedeel word, afhangend van die waardes van δ en ϵ , met twee subgevalle vir elkeen na aanleiding van die waarde van t_1 . Al 8 gevalle word in Figure 8.5(a) tot 8.5(d) gedemonstreer.

As die sinkronisasiefout e in terme van die toevalsveranderlikes t_1 en δ gestip word, word die periodisiteit in Figuur 8.6 sigbaar. Die bogenoemde vier gevalle is dan bloot posisies op die vertikale as, en die 'sub-gevalle' is weerskante van waar die stippellyne in Figuur 8.6 die skuins soliede swart lyne kruis. Na aanleiding van hierdie figuur kan die sinkronisasiefout na die twee gevalle in Vergelyking 8.1 veralgemeen word, mits δ genormaliseer word om in $[\epsilon; \epsilon + T]$ te val en so ook t_1 in $[0; T]$.

$$e = \begin{cases} \delta & \text{vir } \delta < t_1 + \epsilon \\ \delta - T & \text{vir } \delta > t_1 + \epsilon \end{cases} \quad (8.1)$$



Figuur 8.5: Verskillende gevalle van die sinkronisasiefout a.g.v. onakkurate PPS-flanke en asinkrone versyferaars. Die blou strepies dui die monstertydstippe van versyferaar 1 aan, en die oranje strepies dié van versyferaar 2. Die PPS-flanke t_1 en t_2 word deur stippellyne van die ooreenstemmende kleure aangedui. Die monsters wat vervolgens met ranggetal 0 bestempel word, word met vierkante aangedui. Die sinkronisasiefout word met 'n pyl aangedui.



Figuur 8.6: Vir konstante waardes van ϵ en T is die sinkronisasiefout periodies wanneer die toevalsveranderlikes δ en t_1 saam beskou word. As δ genormaliseer word om in $[\epsilon; \epsilon + T]$ te val en so ook t_1 in $[0; T]$, kan die sinkronisasiefout na twee gevalle veralgemeen word, naamlik $e = \delta$ en $e = \delta - T$. Die waarde van die sinkronisasiefout lê in $[\epsilon - T; \epsilon + T]$.

Die sinkronisasiefout lê dan in $[\epsilon - T; \epsilon + T]$. Selfs met perfekte sinkronisasie van **PPS**-flanke, sodat $\epsilon = 0$, is die maksimum sinkronisasiefout dus gelyk aan die monsterperiode. Vir hierdie projek is $\epsilon < 54 \text{ ns}$ en $T = \frac{1}{4800} \text{ s}$, sodat $e \in [-208,2793 \mu\text{s}; 208,3873 \mu\text{s}]$. As die mikrofone 100 m uitmekaar is, en klank beweeg teen $343 \text{ m} \cdot \text{s}^{-1}$, veroorsaak hierdie sinkronisasiefout 'n hoekfout van minstens $0,409^\circ$, volgens Vergelyking 4.1. Dit stem ooreen met 'n posisiefout van minstens 71,5 cm vir klankbronne wat 1 km ver is.

8.7 Voorstelle vir verbetering

Indien die onwaarskynlike maar tog moontlike enkelmonstervertraging steeds onaanvaarbaar sou wees, is daar alternatiewe oplossings wat ondersoek kan word. Hierdie oplossings verhoog egter die kompleksiteit van die stelsel aansienlik:

- 'n Spanningsbeheerde ossilator kan gebruik word as klok vir die versyferaar. Hierdie ossilator word dan in 'n **fasegesinkroniseerde beheerlus (PLL, Phase-Locked Loop)** geplaas wat seker maak dat daar presies 4800 ossilasies per **PPS** plaasvind. Die nadeel hiervan is egter dat die versyferaar geen monsters sal versamel as die **GPS** nog nie die ligging en tyd vasgestel het nie.

- 'n Aparte mikroverwerker kan gebruik word om die klok vir die versyferaar te genereer, eerder as 'n kristal. Hierdie mikroverwerker tree dus soos 'n digitale **PLL** op. Die mikroverwerker kan ook die klok tydelik aan die gang hou in die afwesigheid van betroubare **PPS**-seine.

Al die komponente van die klankopnemer is nou uiteengesit. Die volgende hoofstukke handel oor die ontwikkeling van algoritmes om klankbronne op te spoor, met behulp van opnames wat met hierdie toestel gemaak is.

Hoofstuk 9

Onderstel

Om die komponente wat in §3 tot 8 ontwerp is in die veld te gebruik, moet 'n onderstel ontwerp word waarop alles gemonteer kan word.

Vir die doeleindes van hierdie prototipe is dit nie nodig om 'n weerbestande houer daar te stel nie. As die projek in die toekoms deel sou word van 'n langtermyn-installasie in 'n regte olifanthabitat, sal dit in elk geval nie net weerbestand nie, maar ook olifantbestand gemaak moet word.

Die onderstel word in AutoDesk Inventor ontwerp¹ en met 'n Makerbot Replicator Z18 gedruk. Die gebruik van 'n 3D-drukker bring sekere beperkings teweeg in terme van die oriëntasie waarin elke deel gedruk word. Die onderskeie dele en samestelling van die onderstel word vervolgens kortliks bespreek.

9.1 Batteryhouer

Daar is gekies om standaard 151 mm × 65 mm × 93 mm loodsuurbatterye te gebruik. Dit is die swaarste deel van die apparaat en word dus heel onder geplaas vir stabiliteit. Figuur 9.1 is die ortografiese tekening van hierdie deel van die onderstel.

9.2 Houer vir stroombaanborde en skakelaars

'n Houer om die stroombaanborde en skakelaars te huisves word bo-op die batteryhouer geplaas en met epoksiegom bevestig. Figuur 9.1 is die ortografiese tekening van hierdie deel van die onderstel. Figuur 9.3 toon die houer nadat die komponente geïnstalleer is.

9.3 Mikrofoonarms

In §4.5 is daar besluit op 'n sirkelvormige mikrofoonskikking met 6 mikrofone en 'n radius van 20 cm. Dit word bewerkstellig deur 6 arms (Figuur 9.4) aan die boonste rand van die onderstel te bevestig.

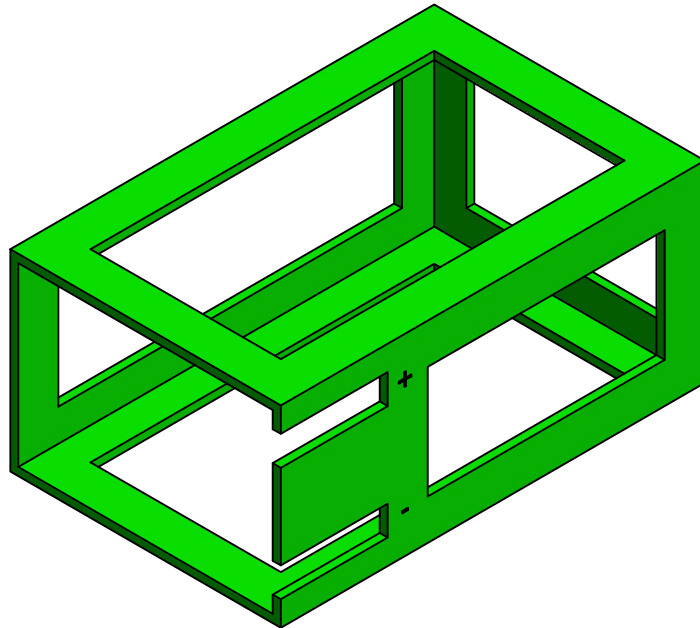
Spesiale koppelaars (Figure 9.5) is ontwerp en gedruk om te verseker dat die arms teen die korrekte hoek en afstand bevestig word sodat die mikrofone reg gespasieer is.

Figuur 9.6 toon die onderstel met die bevestigde mikrofoonarms.

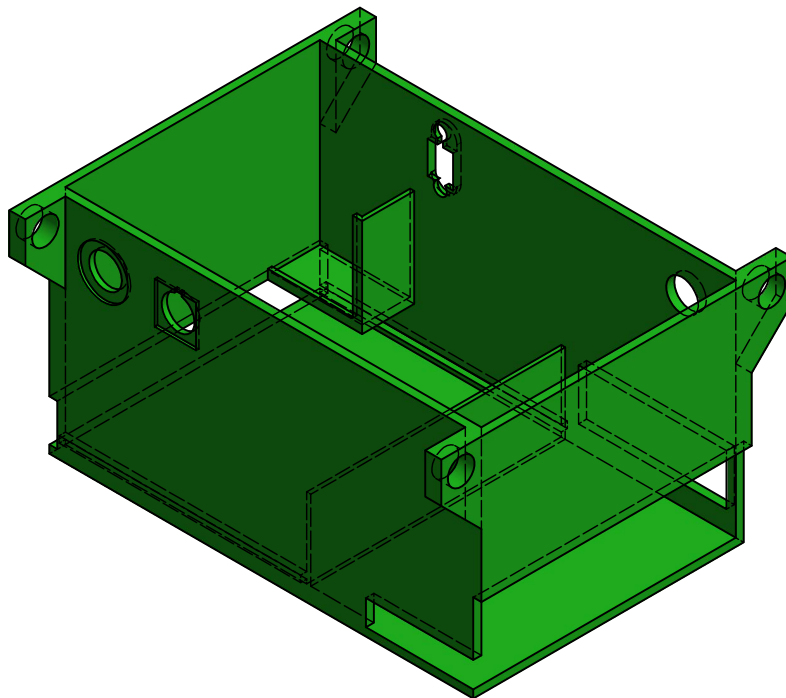
9.4 Handvatsel

Aan elke hoek van die boonste rand van die onderstel is daar 'n gat in die plastiek. 'n Stuk nylontou word daardeur geryg en 'n tuinslang word om die tou gesit om 'n handvatsel te vorm.

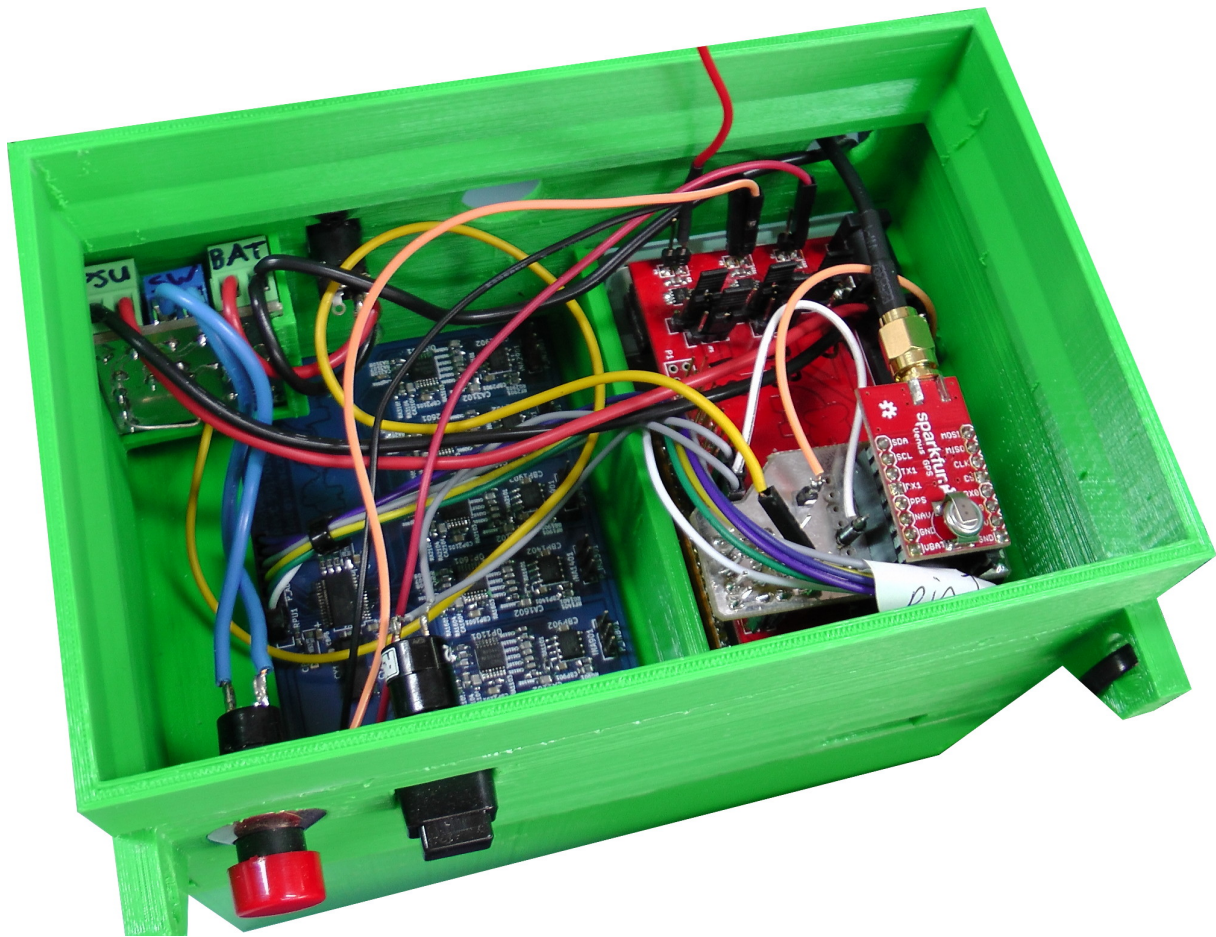
¹ Die AutoDesk Inventor lêers is op DVD E.1 in Bylae E beskikbaar.



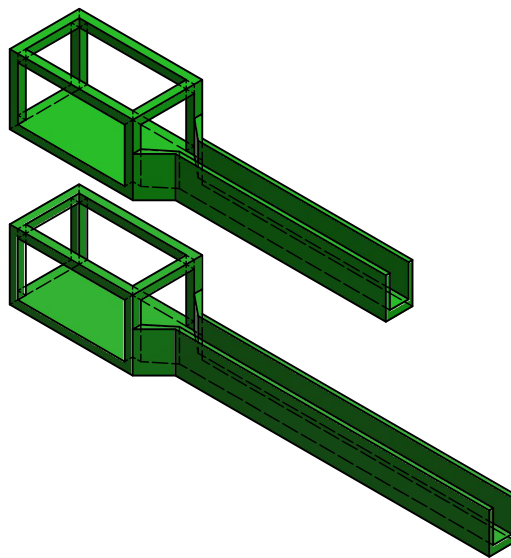
Figuur 9.1: Die onderste deel van die onderstel wat die battery huisves. Die houer het gate om materiaal te spaar en die drukproses te versnel. Skaal: 1:2.



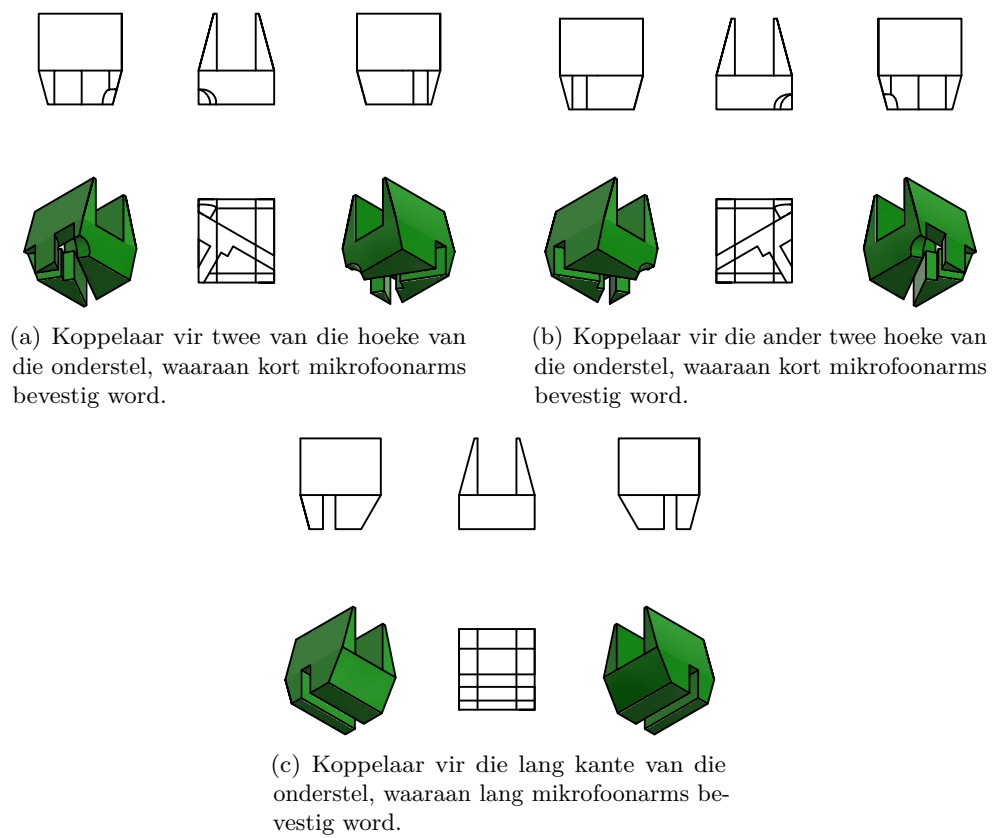
Figuur 9.2: Die boonste deel van die onderstel, waar die al die stroombaanborde gehuisves word. Skaal: 1:2.



Figuur 9.3: Die boonste en onderste dele van die boks is met epoksiegom aan mekaar vasgemaak. Die **RPI** (§7), bufferstroombaanbord (§6), kragbronbord (§3.4) en **Venus638FLPx GPS**-bordjie (§8) is regs te sien. Links bo is die stroombaan wat die kragtoevoer beheer (§3.5). Die blou bord op die bodem is die oudiokoppelvlak (§5). Die rooi skakelaar skakel die apparaat aan (SW1 in Figuur 3.11) en die swart skakelaar versoek die **RPI** om die bedryfstelsel te stop, waarna die apparaat afskakel.



Figuur 9.4: Die mikrofoonarms, wat in twee verskillende lengtes vervaardig word. Twee kort arms word aan die middel van die twee lang kante van die onderstel bevestig, en vier kort arms word elk aan 'n hoek van die onderstel bevestig. Skaal: 1:2.



Figuur 9.5: Koppelaars wat die mikrofoonarms aan die onderstel bevestig. Skaal: 1:2.



Figuur 9.6: Die volledige onderstel met mikrofoonarms en komponente, maar sonder die mikrofone. Bo-op hierdie onderstel word die grondvlak van die **GPS**-met binddraad vasgemaak (sien Figuur 8.3). Dit dien dus ook as 'n deksel.

Hoofstuk 10

Klankbronopsporing

Nou dat die klankopnemer volledig ontwerp is, kan daar aandag gegee word aan algoritmes om die versamelde data te analiseer. In §10.1–10.4 word ’n algoritme ontwikkel wat vir elke opnemer ’n klankbronrigting beraam. Daarna word die rigtingberamings saamgevoeg met behulp van triangulasie (§10.2). In §10.5 word ’n algoritme bespreek wat van alle opnemers gelyktydig gebruik maak en soortgelyk aan **GPS** funksioneer.

10.1 Rigtingberaming met ’n enkele mikrofoonpaar

As twee mikrofone by verskillende posisies dieselfde klankbron waarneem, kan die tydverskil tussen die twee waargenome seine gebruik word om die rigting waaruit die klank aangekom het, te bereken. Figuur 10.1 demonstreer hoe dit moontlik is in die geval waar die klankbron ver van die mikrofone (in die **verrevelde**) is. As die tydverskil beraam kan word, is afstand b ook bekend en kan die rigting α met behulp van Vergelyking 10.1 bepaal word.

$$\alpha = \arcsin\left(\frac{\tau v}{d}\right)$$

waar α = rigting van klankbron [$^\circ$],
 τ = tydvertraging [s],
 v = spoed van klank = $343 \text{ m} \cdot \text{s}^{-1}$,
 en d = afstand tussen mikrofone [m].

(10.1)

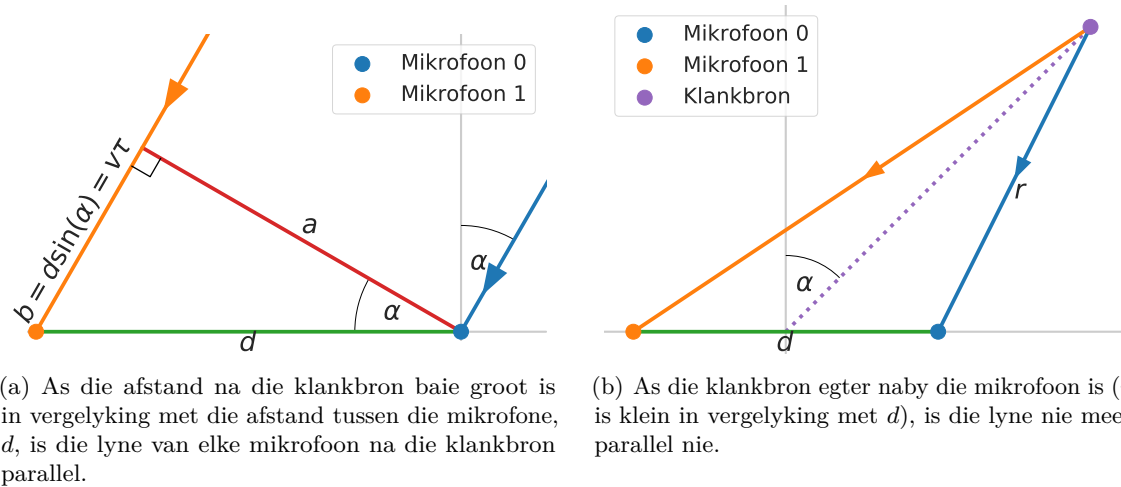
Daar is egter ’n dubbelsinnigheid in die resultaat α , omdat ’n klankbron wat om die horisontale as van Figuur 10.1 gekeer word, dieselfde tydvertraging τ sal veroorsaak. Hierdie dubbelsinnigheid word in §10.3 opgelos, wanneer meer as een **mikrofoonpaar** saam gebruik word. Die res van hierdie afdeling werk met die aanname dat die klankbron tussen 0° en 180° lê.

Wanneer die klankbron in die **nabye veld** is, veroorsaak die **verrevelde**-aanname ’n hoekfout. Figuur 10.2(a) toon die fout vir verskeie klankbronposisies. Voortaan word daar aangeneem dat die klankbronne slegs in die **verrevelde** voorkom.

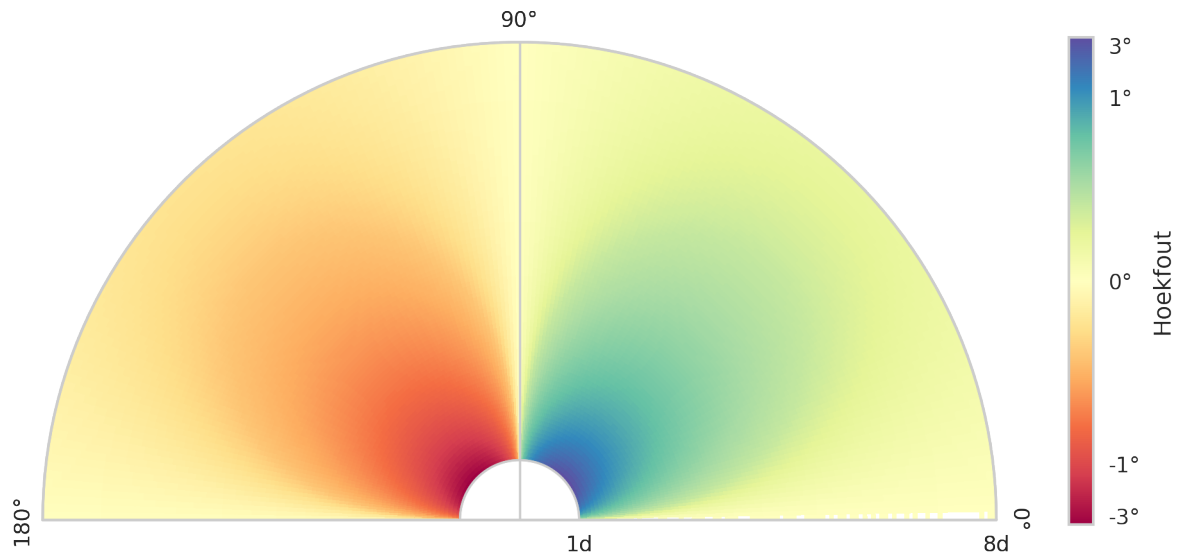
Daar bestaan ook ’n verwantskap tussen die akkuraatheid waarmee τ bepaal kan word en die resolusie van α . Om die algoritmes in hierdie hoofstuk te simuleer, moet bronseine opgewek word en die beweging van daardie seine deur die atmosfeer tot by die mikrofone gesimuleer word. Dit is egter nie moontlik om kontinue seine waarvoor daar nie geslote uitdrukkings bestaan, op ’n rekenaar te simuleer nie. Die seine moet dus teen ’n bepaalde monstertempo opgewek word.

Figuur 10.3 toon die verwantskap van Vergelyking 10.1 vir drie verskillende monstertempos. Hoe laer die monstertempo, hoe laer is die resolusie van τ . Omdat \arcsin nie ’n lineêre funksie is nie, is die resolusie van α nie konstant nie.

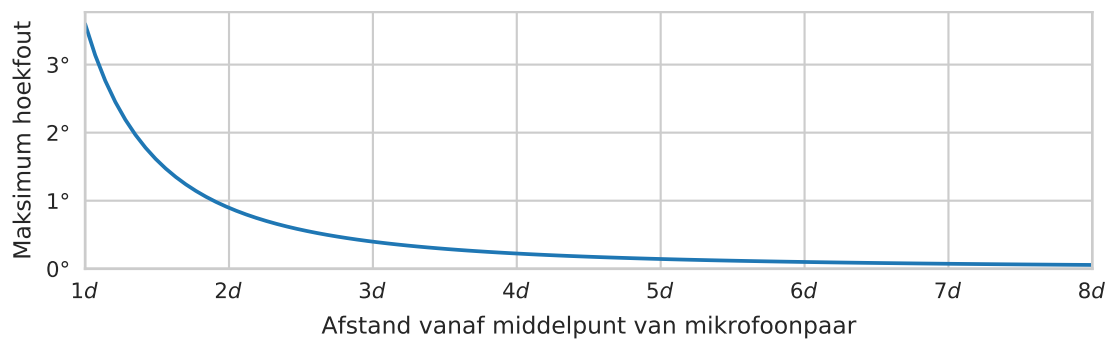
Die minimum en maksimum resolusie van α word in Figuur 10.4 teenoor die monstertempo gestip. Hieruit is dit duidelik dat die simulaties teen ’n baie hoër monstertempo uitgevoer



Figuur 10.1: Hoekberekening vir klankbronne wat ver of naby is. Die blou en oranje lyne stel die voortplantingsrigting van klankgolwe voor.

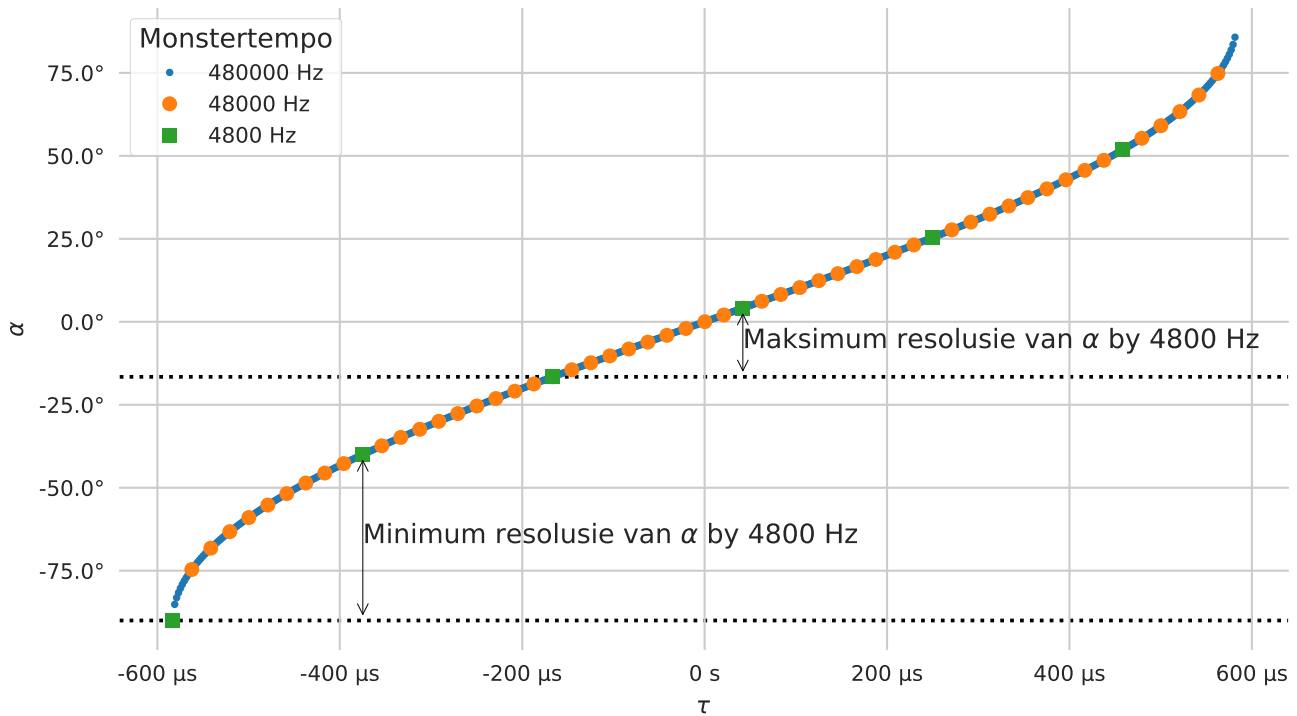


(a) Die hoekfout in terme van die klankbronposisie. 'n Logaritmiese kleurskaal word gebruik om die data duideliker voor te stel. Die maksimum hoekfout is altyd by $\pm 45^\circ$ en neig na 'n kleiner waarde van α .

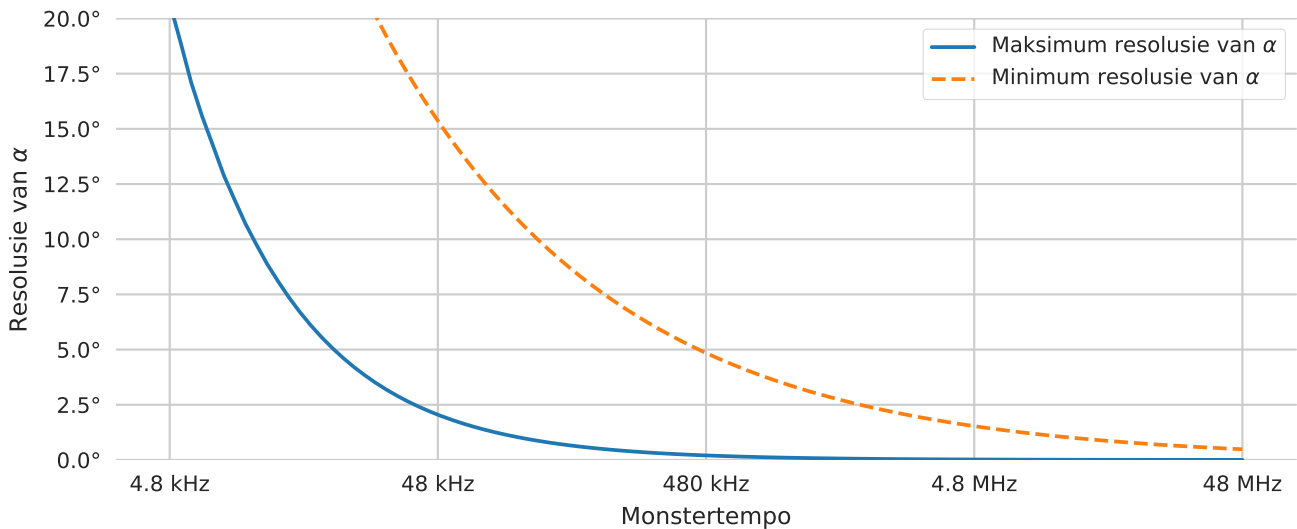


(b) Die maksimum hoekfout teenoor die klankbronaafstand.

Figuur 10.2: Die hoekfout wat deur die *verrevelde*-aanname veroorsaak word as die klankbron naby die mikrofoonpaar is. Die mikrofone is by $(\frac{d}{2}; 0)$ en $(-\frac{d}{2}; 0)$ in Cartesiese koördinate, en alle afstande is in terme van d .



Figuur 10.3: Die verwantskap tussen τ en α in Vergelyking 10.1 vir verskillende monstertempos, met $d = 0,2 \text{ m}$ en $v = 343 \text{ m} \cdot \text{s}^{-1}$.



Figuur 10.4: Die maksimum en minimum resoluise van α in Vergelyking 10.1 teenoor monstertempo, met $d = 0,2 \text{ m}$ en $v = 343 \text{ m} \cdot \text{s}^{-1}$. Hoe hoër die monstertempo, hoe beter is die resoluise van α . Dit het slegs 'n invloed op simulaties, omdat seine in praktyk kontinu is.

moet word as die monstertempo van die apparaat se versyferaar. Om dit te doen, sal daar van 100-voudige **oormonstering** gebruik gemaak word.

10.1.1 Kruiskorrelasie

Om die klankbronrigting te bepaal, moet die tydvertraging τ eers beraam word. Die posisie van die kruin van die kruiskorrelasie $k_{s_1, s_2}[i]$ tussen die seine $s_1[n]$ en $s_2[n]$ van die twee mikrone gee 'n goeie benadering van τ . Diskrete kruiskorrelasie word deur Vergelyking 10.2 gedefinieer. $k_{s_1, s_2}[i]$ gee dus die mate van ooreenstemming tussen seine s_1 en s_2 wanneer sein s_1 met i monsters vertraag word. Daar word aangeneem dat s_1 en s_2 nul is buite die **definisieversameling**.

$$k_{s_1, s_2}[i] = \sum_{n=-\infty}^{\infty} s_1[n] s_2[n-i] \quad (10.2)$$

Die diskrete kruiskorrelasie is soortgelyk aan die diskrete **vouprodukt** en kan dus met behulp van Fourier-transforms bereken word, soos aangetoon in Vergelyking 10.3.

$$\begin{aligned} s_1[i] * s_2[i] &= \sum_{n=-\infty}^{\infty} s_1[n] s_2[i-n] \\ \therefore k_{s_1, s_2}[i] &= s_1[i] * s_2[-i] \\ \therefore \mathcal{F}\{k_{s_1, s_2}[i]\} &= \mathcal{F}\{s_1[i] * s_2[-i]\} \\ &= \mathcal{F}\{s_1[i]\} \mathcal{F}\{s_2[-i]\} \\ \therefore k_{s_1, s_2}[i] &= \mathcal{F}^{-1}\{\mathcal{F}\{s_1[i]\} \mathcal{F}\{s_2[-i]\}\} \end{aligned} \quad (10.3)$$

Om die kruiskorrelasie met **diskrete Fourier-transforms (DFTs)** te implementeer, moet beide seine met genoeg¹ **nulle gestop** word om oorvleueling te voorkom voordat die **DFTs** geneem word. Dieselfde aantal nulle kan weer van die resultaat van die inverse **DFT** afgesny word sodat die indekse van die resultaat reg oplyn.

'n Eerste beraming vir die werklike tydverskil tussen seine s_1 en s_2 word deur Vergelyking 10.4 gegee.

$$\hat{\tau} = T \arg \max_i (k_{s_1, s_2}[i]) \quad (10.4)$$

10.1.2 Interpolasie

Soos genoem in §4.2, is die resolusie van die kruiskorrelasie van twee diskrete seine gelyk aan die monsterperiode, $T = 208,3 \mu\text{s}$. Volgens Vergelyking 10.1 stem dit met 'n maksimum resolusie van hoogstens $20,9^\circ$ in α ooreen, wat nie goed genoeg is nie.

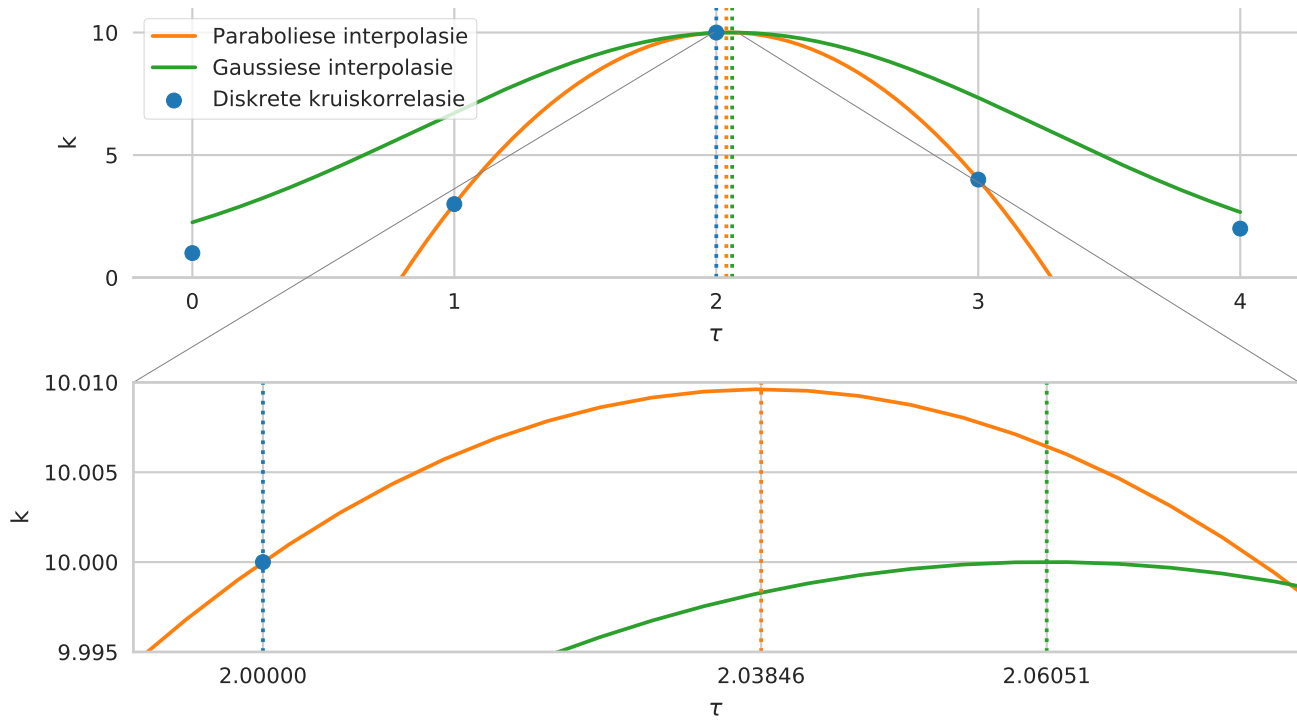
Dit is egter moontlik om die waardes van die kruiskorrelasie naby $\hat{\tau}$ te interpoleer om 'n toppunt tussenin twee monsters te verkry. Die kontinue (geïnterpoleerde) kruiskorrelasie in terme van die tydvertraging $\tau = Ti$ word as $k_{s_1, s_2}(\tau) = k(\tau)$ geskryf (die voetskrif s_1, s_2 word voortaan ter wille van bondigheid uitgelaat). Die ware tydverskil word dan deur Vergelyking 10.5 gegee.

$$\tau_{pk} = \arg \max_{\tau} (k(\tau)) \quad (10.5)$$

Twee metodes van interpolasie, naamlik a) paraboliese en b) Gaussiese interpolasie is ondersoek. Figuur 10.5 toon voorbeelde van beide interpolasies.

Paraboliese interpolasie kan met behulp van 'n geslote oplossing geïmplementeer word, wat dus baie vinnig is om te bereken. Vir 'n parabool $k = a\tau^2 + b\tau + c$ met drie bekende punte

¹ Die som van die lengte van beide seine, minus 1.



Figuur 10.5: 'n Voorbeeld van paraboliese en Gaussiese interpolasie van 3 punte van 'n kruiskorrelasieresultaat. 'n Baie klein verskil in die posisie van hierdie toppe maak 'n groot verskil aan α in Vergelyking 10.1.

$(\tau_1; k_1)$, $(\tau_2; k_2)$ en $(\tau_3; k_3)$ kan Vergelyking 10.6 gebruik word om die kruin by $(\tau_{pk}; k_{pk})$ te vind².

$$\begin{aligned} a &= \frac{\tau_3(k_2 - k_1) + \tau_2(k_1 - k_3) + \tau_1(k_3 - k_2)}{(\tau_1 - \tau_2)(\tau_1 - \tau_3)(\tau_2 - \tau_3)}, \\ b &= \frac{\tau_3^2(k_1 - k_2) + \tau_2^2(k_3 - k_1) + \tau_1^2(k_2 - k_3)}{(\tau_1 - \tau_2)(\tau_1 - \tau_3)(\tau_2 - \tau_3)}, \\ c &= \frac{\tau_2\tau_3k_1(\tau_2 - \tau_3) + \tau_1\tau_3k_2(\tau_3 - \tau_1) + \tau_1\tau_2k_3(\tau_1 - \tau_2)}{(\tau_1 - \tau_2)(\tau_1 - \tau_3)(\tau_2 - \tau_3)}, \end{aligned} \quad (10.6)$$

waar $\tau_2 = \hat{\tau} = T \arg \max_i (k_{s_1, s_2} [i])$ en $k_2 = \max (k_{s_1, s_2} [i])$.

Dan is $\tau_{pk} = \arg \max_{\tau} (k(\tau)) = \frac{-b}{2a}$ en $k_{pk} = \max (k(\tau)) = c - \frac{b^2}{4a}$.

Interpolasie met behulp van 'n Gaussiese **kerndigtheidsberaming** is oorspronklik deur Zhang en Wu (2006) voorgestel. 'n **Kerndigtheidsberaming** is die som van verskeie waarskynlikheidsverdelings, waar elke verdeling dieselfde vorm het en elke verdeling om 'n ander observasie gesentreer is. Die verdeling wat gebruik word, word die 'kern' genoem, en is in hierdie geval 'n Gaussiese verdeling met 'n vasgestelde standaardafwyking σ .

N punte rondom die eerste beraming van τ (Vergelyking 10.4) word as die observasies in die **kerndigtheidsberaming** gebruik. Vergelyking 10.7 definieer hierdie **kerndigtheidsberaming** en gee

² Die formules vir a , b en c in Vergelyking 10.6 kan afgelei word deur die vergelyking $\begin{bmatrix} k_1 \\ k_2 \\ k_3 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \begin{bmatrix} \tau_1^2 & \tau_1 & 1 \\ \tau_2^2 & \tau_2 & 1 \\ \tau_3^2 & \tau_3 & 1 \end{bmatrix}$ met behulp van **matriksomkering** op te los.

die eerste twee afgeleides daarvan.

$$f(\tau|\sigma) = \sum_{n=1}^N y_n \frac{1}{\sqrt{2\pi} \sigma} \exp \left[-\frac{(\tau - \tau_n)^2}{2\sigma^2} \right] \quad (10.7a)$$

$$f'(\tau|\sigma) = \sum_{n=1}^N y_n \frac{\tau_n - \tau}{\sqrt{2\pi} \sigma^3} \exp \left[-\frac{(\tau - \tau_n)^2}{2\sigma^2} \right] \quad (10.7b)$$

$$f''(\tau|\sigma) = \sum_{n=1}^N y_n \frac{(\tau_n - \tau)^2 - \sigma^2}{\sqrt{2\pi} \sigma^5} \exp \left[-\frac{(\tau - \tau_n)^2}{2\sigma^2} \right] \quad (10.7c)$$

waar τ = kontinue tydvertraging,

$(\tau_n; y_n)$ = punte in die kruiskorrelasieresultaat naaste aan die kruin,

N = die aantal punte wat gebruik word,

en σ = die standaardafwyking van die Gaussiese kern

sodat $\tau_{pk} = \arg \max_{\tau} (k(\tau)) \approx \arg \max_{\tau} (f(\tau|\sigma))$.

Vir die gegewe waardes van N en σ kan 'n optimeringsfunksie gebruik word om die waarde van τ te vind waarvoor $f(\tau|\sigma)$ 'n maksimum aanneem. Die oplossing sal binne een monsterperiode van $\hat{\tau}$ lê en behoort 'n goeie benadering vir die ware tydverskil te wees.

Alhoewel die Gaussiese interpolasie effens stadiger as die paraboliese interpolasie is, bied dit beter akkuraatheid (sien Figuur 10.6 in §10.1.4). Die Gaussiese metode is gekies.

10.1.3 Vind die piek van die geïnterpoleerde kruiskorrelasie

Vir hierdie projek is die **vertrouensgebied toegevoegde-gradiënt Newton (T-NCG, Trust-region Newton Conjugate Gradient)** optimeringstegniek gekies om $\arg \max_{\tau} (f(\tau|\sigma))$ te vind. Hierdie tegniek maak gebruik van die eerste en tweede afgeleides (oftewel Jakobiaan en Hessiaan – Vergelykings 10.7b en 10.7c) van die funksie wat geoptimeer word en is baie vinniger as ander metodes wanneer die afgeleides vinnig bereken kan word (Nocedal en Wright, 2006).

In **T-NCG** word daar vir elke iterasie 'n plaaslike kwadratiese model van $f(\tau|\sigma)$ gebou, waarna die oplossing 'n stap langs τ in die rigting van die optimum τ_{pk} neem. Die kwadratiese model is slegs geldig vir 'n klein gebied rondom die beginpunt van elke iterasie. Die konsep '**vertrouensgebied**' verwys na die omvang in τ waarvoor daar aangeneem word dat die kwadratiese model 'n goeie benadering vir $f(\tau|\sigma)$ is. Die **vertrouensgebied** bepaal die maksimum toelaatbare stap in τ wat die oplosser tydens elke iterasie mag maak. Na elke sprong word 'n nuwe **vertrouensgebied** bereken. Dit verskil dus van die gewone **toegevoegde-gradiënt Newton (NCG, Newton Conjugate Gradient)** deurdat die stappe beperk word. Hierdie beperking dien as voorsorgmaatreël om seker te maak dat die stappe nie so groot is dat die oplosser in die eerste stap ver verby die piek beweeg nie.

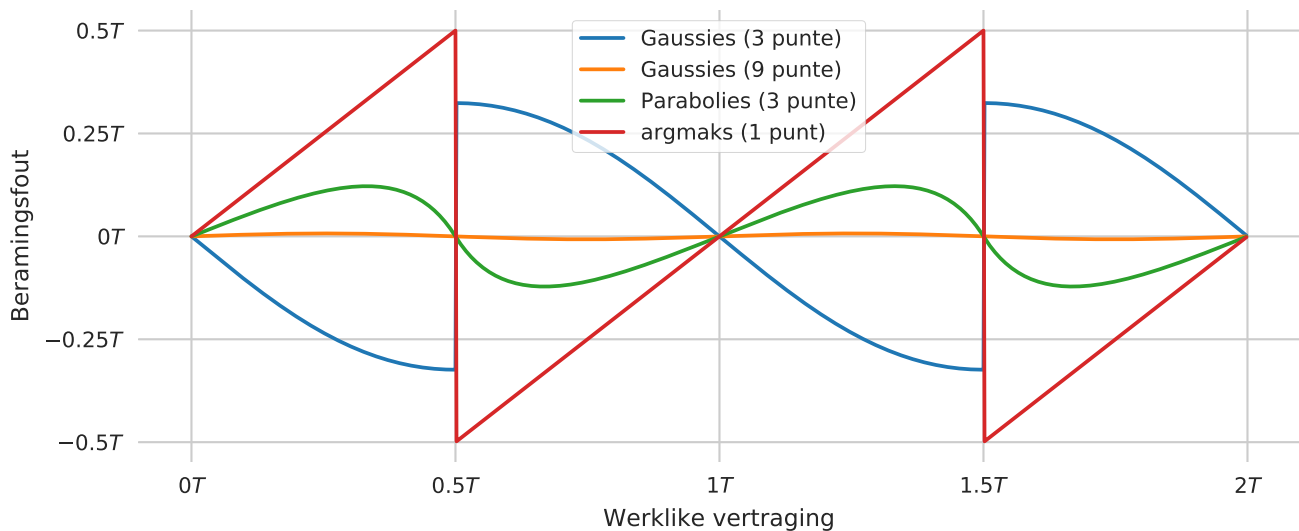
Die **Scipy**-biblioteek bied 'n implementasie van **T-NCG** (Verskeie outeurs, 2016). Omdat hierdie implementasie slegs minimering doen, moet die tekens van Vergelyking 10.7 omgeruil word. Die minimeringsfunksie **scipy.optimize.minimize** word met die volgende argumente³ geroep om hierdie optimering uit te voer:

fun Die funksie⁴ om te minimeer: $-f(\tau|\sigma)$.

x0 'n Eerste benadering vir die posisie van die minimum: $\hat{\tau}$.

³ Die name van die argumente word deur die SciPy-biblioteek bepaal.

⁴ Die wiskundige funksies f , f' en f'' is as Python-funksies geïmplementeer wat vir elke **float-inset**waarde τ 'n **float** teruggee. In Python is dit normale praktyk om funksies as argumente aan ander funksies deur te gee.



Figuur 10.6: 'n Vergelyking van beramingsfoute met verskillende wyses van interpolasie. 'Argmaks' is die eerste benadering wat in Vergelyking 10.4 gegee is. Die passing van 'n parabool op drie punte bied goeie resultate. Die Gaussiese metode bied die heel beste resultate, mits meerdere punte gebruik word.

`method 'trust-ncg'`

`jac` Die eerste afgeleide van die funksie wat geminimeer word: $-f'(\tau|\sigma)$.

`hess` Die tweede afgeleide van die funksie wat geminimeer word: $-f''(\tau|\sigma)$.

`options`

`gtol` Die gewenste akkuraatheid van die oplossing: 1×10^{-8}

`max_trust_radius` Die maksimum **vertrouensgebied**. Dit beperk die maksimum stap in τ wat by elke iterasie geneem kan word. Die ware oplossing lê altyd binne een periode van $\hat{\tau}$, dus: T

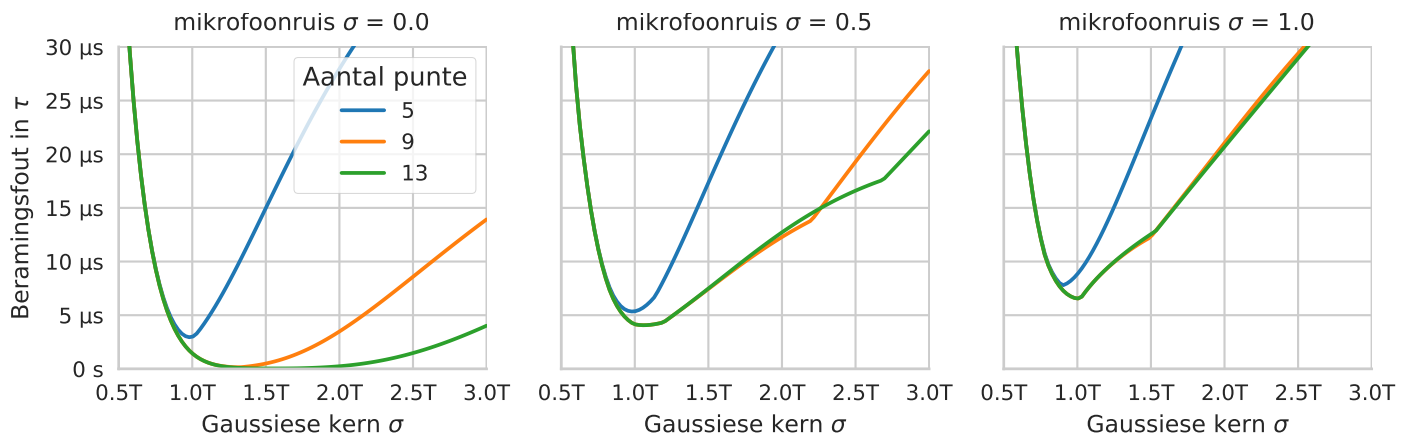
`initial_trust_radius` Die aanvanklike **vertrouensgebied**. Dit is die maksimum stap in τ tydens die eerste iterasie: $0,5T$

10.1.4 Simulasies

Die Gaussiese metode het twee parameters, wat die beraming beïnvloed, naamlik die standaardafwyking van die Gaussiese kern σ en die aantal kruiskorrelasiepunte N weerskante van die 'argmaks' wat in die **kerndigtheidsberaming** gebruik word. Simulasies is uitgevoer om te bepaal watter kombinasie van parameters die kleinste beramingsfoute sal gee.

Die eerste simulatie vergelyk paraboliese en Gaussiese interpolasie by verskeie waardes van N . Figuur 10.6 vergelyk die beramingsfoute, d.i. die verskil tussen die beraamde en ware tydvertraging van die twee seine met behulp van die onderskeie interpolasiemetodes. As die ware vertraging 'n halwe monsterperiode is, is die parabool van die paraboliese interpolasie presies simmetries, en daarom is die beramingsfout 0. Die ander metodes se beramingsfoute bereik 'n maksimum by 'n halwe monsterperiode. Die Gaussiese interpolasie bied beter resultate mits N groot genoeg is. Al die metodes is eksak by vertraging van heelgetal-veelvoude van die monsterperiode.

Die tweede simulatie toets Gaussiese interpolasie by verskeie waardes van σ en N , wanneer daar verskillende vlakke van ongekorreleerde ruis by een van die seine getel word. Die bronsein wat gegenereer is, is Gaussiese ruis met 'n standaardafwyking van 1. Hierdie sein is met verskeie tydwaardes tussen 0 s en $208,3 \mu\text{s}$ vertraag (0 tot 1 monsterperiode teen 4800 Hz) en 'n ander ongekorreleerde ruissein (wat mikrofoonruis voorstel) is bygetel. Daarna is elke vertraagde sein met ruis met die oorspronklike sein gekorreleer. Die kruiskorrelasies is dan gebruik om die



Figuur 10.7: Vergelyking van parameters vir die Gaussiese interpolasie. Meer punte bied beter resultate. As daar ruis by die seine gevoeg word, werk 'n Gaussiese kern met 'n standaardafwyking van T die beste.

vertraging met verskillende parameters te beraam. Figuur 10.7 toon die resultate. As daar geen ruis is nie, werk 'n Gaussiese kern met 'n standaardafwyking van 1,5 keer die monsterperiode goed. Hoe meer punte oor die algemeen gebruik word, hoe beter vaar die Gaussiese beraming. As daar ruis by die seine gevoeg word, bied 'n Gaussiese kern met 'n standaardafwyking van presies 1 monsterperiode egter die beste beramings. Hierdie tegniek werk dus goed, selfs wanneer die ongekorreleerde ruis dieselfde amplitude as die bronsein bereik.

Derdens is klankbronrigtingberaming met 'n **mikrofoonpaar** vir 'n spesifieke klankbronposisie gesimuleer:

1. $n \in \{0, 1\} \triangleq$ mikrofoonindeks
2. Plaas mikrofone 20 cm uitmekaar, by $(0,1 \text{ m}; 0 \text{ m})$ en $(-0,1 \text{ m}; 0 \text{ m})$ met 'n standaardafwyking van σ_{mp} op die posisie van elkeen.
3. Simuleer klankgolfvoortplanting en mikrofone:
 - 3.1. $s_{\text{bron}} \leftarrow$ wek 1 s Gaussiese ruissein met 'n standaardafwyking van $2^{23} - 1$ teen 480 kHz op⁵, en beperk die bandwydte tot 500 Hz. Dit is dus 'n sein waarvan die amplitude 'n standaardafwyking het wat die versyferaar sou laat afsny as die klankbron reg langs die mikrofoon was. Die 500 Hz bandwydte stem ongeveer ooreen met dié van 'n olifantdreungeluid.
 - 3.2. $P_n \triangleq$ vertraging en verswakking van bron na mikrofoon n om klankgolfvoortplanting in lug te simuleer. Die spoed van klank word as 'n konstante $343 \text{ m} \cdot \text{s}^{-1}$ beskou. Die sein word verswak deur dit deur die kwadraat van die afstand in meter te deel. Die sein word met 'n heelgetal aantal monsters vertraag, waarna dit tot 4800 Hz **uitgedun** word.
 - 3.3. $H_n \triangleq$ hoogdeurlaatfilter vir mikrofoon n (soos beskryf in §4.2).
 - 3.4. $\sigma_r \leftarrow 2^{N_{\text{ruisbisse}} - 1}$, waar $N_{\text{ruisbisse}}$ die aantal ruiserige bisse⁶ in die versyferde sein

⁵ Die standaardafwyking van hierdie sein is dus ongeveer so groot soos die maksimum amplitude van 'n sein wat deur die 24-bis versyferaar gemeet kan word. Daar is egter nie vasgestelde maksimum amplitude nie, want die monsters is normaal verdeel.

⁶ Ware Gaussiese ruis het nie 'n maksimum amplitude nie. Die 'aantal ruiserige bisse' in 'n klankstelsel word gewoonlik gedefinieer as die aantal bisse wat nodig is om die gemiddelde ruisvlak voor te stel. As P die gemiddelde drywing van die ruis in dBFS is, en M is die maksimumwaarde van die versyferaar (gelyk aan 0 dBFS), dan is die aantal ruiserige bisse gelyk aan $\log_2(M \times 10^{P/20})$. Vir die simulaties word die standaardafwyking van die ruis as die 'drywing' beskou. Solank dieselfde definisie vir die bronseine en steurseine gebruik word, bly dit 'n geldige vergelyking. Dit is immers die **SNR** wat die werksverrigting van die stelsel beïnvloed, nie die absolute amplitude van enige van die betrokke sein nie.

- voorstel.
- 3.5. $s_{\text{steur},n} \leftarrow$ ongekorreleerde Gaussiese ruis by kanaal n met standaardafwyking σ_r (die steursein).
 - 3.6. $s_0 \leftarrow H_0(P_0(s)) + s_{\text{steur},0}$.
 - 3.7. $s_1 \leftarrow H_1(P_1(s)) + s_{\text{steur},1}$.
4. Beraam tydvertraging tussen mikrofoonseine:
- 4.1. $k_{s_0,s_1} \leftarrow$ kruiskorrelasie van s_0 en s_1 .
 - 4.2. $\tau \leftarrow$ posisie van kruin van kruiskorrelasie, met behulp van Gaussiese **kerndigtheidsberaming**.
5. Bereken die rigting:
- 5.1. $\alpha \leftarrow$ klankbronrigting met behulp van Vergelyking 10.1.
 - 5.2. $\alpha_{\text{verw.}} \triangleq$ verwysingshoek van die **mikrofoonpaar**, d.w.s. loodreg op die lyn van mikrofoon 0 na mikrofoon 1 (Sien Figuur 10.1).
 - 5.3. $\phi \leftarrow \alpha_{\text{verw.}} - \alpha =$ klankbronrigting in die poolkoördinaat-sisteem.

Die simulaties word uitgevoer met dieselfde monstertempo as die versyferaar, naamlik 4800 Hz. **Oormonstering** word gebruik om 'n goeie benadering vir 'n kontinue bronsein te kry. Die sein s word dus opgewek met 'n monstertempo van 100×4800 Hz, en die resultate van $H_n(s)$ word met 'n faktor van 100 **uitgedun**. Die resolusie van die gesimuleerde vertraging van die klankbron is dus tussen $0,2^\circ$ en $4,8^\circ$ soos ook in Figuur 10.4 te sien is.

Ses simulaties word uitgevoer:

Ideaal Onder 'ideale' toestande. Daar is geen foute op die mikrofoonposisie nie, die hoogdeurlaatfilters is eksak, en geen steurseine word by die bronsein getel nie.

HDF Met nie-eksakte hoogdeurlaatfilters soos beskryf in §4.2.

MP Met 'n 1 cm standaardafwyking op die mikrofoonposisies. 1 cm is meer as die verwagte maksimum afwyking van die mikrofoonposisies vir die prototipe soos wat dit vir hierdie projek met die hand gebou is.

Ruis2, Ruis4 en Ruis8 Met ongekorreleerde ruis by die mikrofoonseine gevoeg. Die ongekorreleerde ruis stel ruis voor wat in die opnemer voorkom, en is dus konstant m.b.t. die afstand van die klankbron. Dit beperk die effektiewe afstand waaroor die apparaat olifante sal kan opspoor.

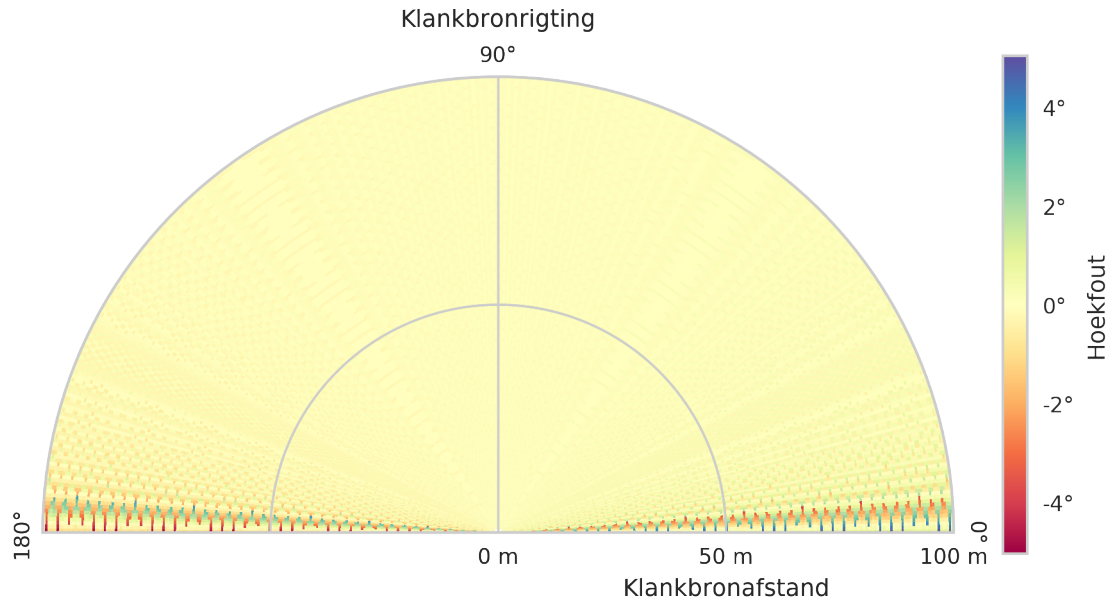
Tabel 10.1 toon die presiese parameters wat in elke simulatie gebruik is. Bronposisies by 200 diskrete stappe van 1 m tot 100 m vanaf die oorsprong en 360 stappe van rigting 0° tot 180° word ondersoek. Vir elke bronposisie word die simulatie $N_{\text{simulasies}}$ keer uitgevoer – elke keer met nuut opgewekte bronseine en steurseine.

Figuur 10.8 toon die gemiddelde rigtingberamingsfout as 'n funksie van die klankbronposisie vir die simulatie onder ideale toestande. Die beramingsfoute naby 0° en 180° is aan die diskretisering van die rigting α en die tydvertraging τ in Vergelyking 10.1 te wyte en bestaan dus slegs in die simulaties en nie in die praktyk nie. Dit dien as verwysingsraamwerk vir die daaropvolgende simulaties. Hierdie beramingsfoute kan so klein gemaak word as wat nodig is deur die mate van oormonstering (tans 100 maal) in die simulaties te verhoog. Simulasies met nog meer oormonstering verg egter 'n onregverdigbare hoeveelheid geheue en verwerkertyd⁷.

⁷ Die simulaties vir Figure 10.8 tot 10.10 het elk ongeveer 55 ure op 'n 8-kern Intel Xeon geneem.

	Ideaal	HDF	MP	Ruis2	Ruis4	Ruis8
H_n	$H_0 = H_1$	$H_0 \approx H_1$, §4.2	$H_0 = H_1$	$H_0 = H_1$	$H_0 = H_1$	$H_0 = H_1$
σ_{mp}	0	0	1 cm	0	0	0
$N_{\text{ruisbisse}}$	0	0	0	2	4	8
$N_{\text{simulasies}}$	100	100	100	30	30	30
Resultate:	Fig. 10.8	Fig. 10.8	Fig. 10.9	Fig. 10.8	Fig. 10.10(a)	Fig. 10.10(b)

Tabel 10.1: Simulasieparameters vir rigtingberaming met 'n enkele mikrofoonpaar.



Figuur 10.8: Die gemiddelde rigtingberamingsfout as 'n funksie van die klankbronposisie vir simulasies sonder ruis. Die beramingsfoute in hierdie simulasie is aan die diskretisering van τ te wyte. Naby 0° en 180° is die gemiddelde fout hoër, as gevolg van die hoë helling van arcsin teenoor τ saam met die diskretisering van τ , soos aangedui in Figure 10.3 en 10.4. Die resultate van die simulasie met 'n realistiese hoogdeurlaatfilter lyk identies aan hierdie figuur. Die faseverskille wat in §4.2 bespreek is, het dus nie 'n groot impak op die akkuraatheid van die rigtingberaming met breë-band seine nie.

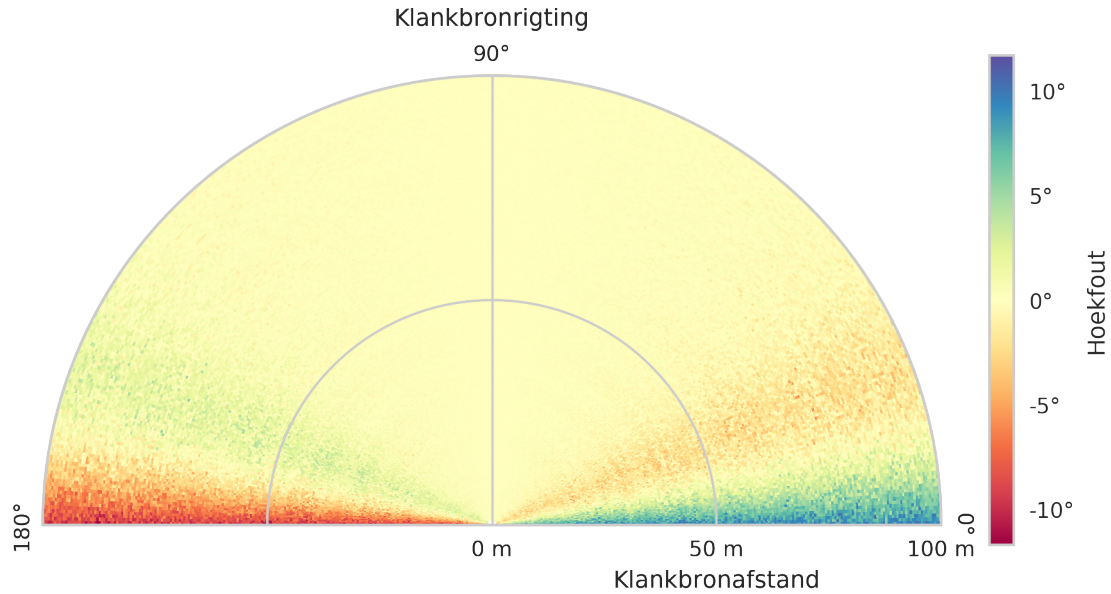
Die resultate van die simulasie met 'n nie-ideale hoogdeurlaatfilter lyk identies aan die simulasie onder ideale toestande, en daarom word geen Figuur getoon nie. Die komponent-toleransies van die ontwerpte hoogdeurlaatfilter is dus voldoende. Die simulasie met onakkurate mikrofoonposisies toon 'n verhoging in beramingsfoute naby 0° en 180° en is in Figuur 10.9 te sien.

10.2 Triangulasie van rigtingberamings

As die klankbron in die **nabye veld** is ten opsigte van twee of meer **mikrofoonskikkings**, maar in die **verre veld** ten opsigte van die mikrofone in elke skikking, kan die rigtingberamings (soos in §10.1 bereken) gebruik word om die klankbronposisie vas te stel.

Elke **mikrofoonskikking** kan voorgestel word as 'n nodus met middelpunt $\mathbf{q}_n = [q_{nx}, q_{ny}]$ en rigtingberaming ϕ_n , waar $n \in \{1, 2, \dots, N\}$. As die klankbronposisie deur koördinaat $\mathbf{p} = [p_x, p_y]$ voorgestel word, word die verband tussen die posisie van die **mikrofoonskikking**, die rigtingberaming en die klankbronposisie deur Vergelyking 10.8 gegee.

$$\tan(\phi_n) = \frac{p_y - q_{ny}}{p_x - q_{nx}} \quad (10.8)$$



Figuur 10.9: Die gemiddelde rigtingberamingsfout as 'n funksie van die klankbronposisie vir 'n simulase met mikrofoonposisiefoute met 'n standaardafwyking van 1 cm. Foute in die mikrofoonposisies veroorsaak 'n verhoging in beramingsfoute naby 0° en 180° as dit met die ideale toestande in Figuur 10.8 vergelyk word.

Met meer as twee skikkings is die stelsel oorgedetermineer (Vergelyking 10.9) en kan 'n oplossing vir die klankbronposisie met die **kleinstekwadratemetode** verkry word.

$$\begin{bmatrix} -\tan(\phi_1) & 1 \\ -\tan(\phi_2) & 1 \\ \vdots & 1 \\ -\tan(\phi_N) & 1 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} q_{1,y} - q_{1,x} \tan(\phi_1) \\ q_{2,y} - q_{2,x} \tan(\phi_2) \\ \vdots \\ q_{N,y} - q_{N,x} \tan(\phi_N) \end{bmatrix} \quad (10.9)$$

Hierdie metode van triangulasie kan natuurlik ook op verskeie **mikrofoonpare** toegepas word, omdat 'n **mikrofoonpaar** maar net 'n lineêre **mikrofoonskikking** met twee elemente is.

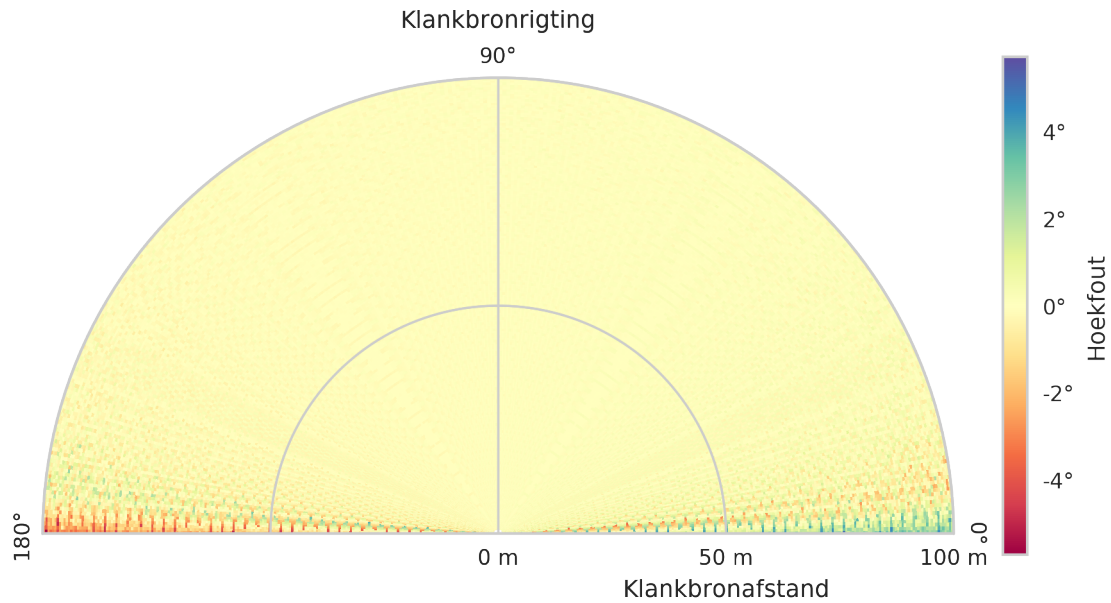
10.3 Hantering van hoekdubbelsinnigheid

Die dubbelsinnigheid in die klankbronrigtingberaming met 'n enkele **mikrofoonpaar** word in Figuur 10.11 uitgebeeld. Hierdie onsekerheid bestaan nie net vir **mikrofoonpare** nie, maar vir enige lineêre **mikrofoonskikking**.

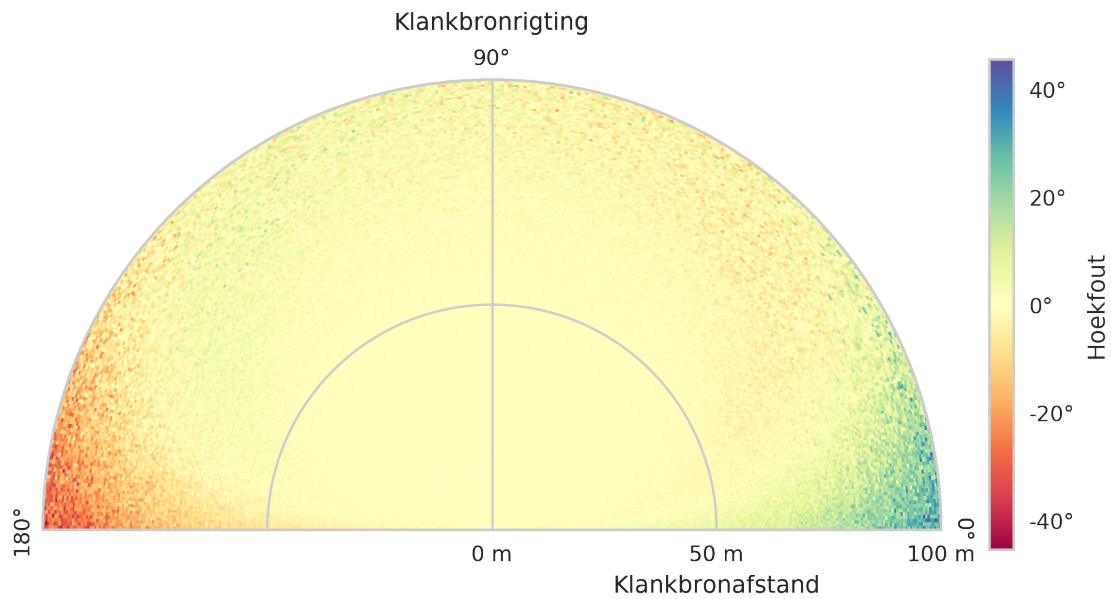
Ter wille van bondigheid word daar voortaan na 'n **mikrofoonpaar** of **mikrofoonskikking** saam met sy rigtingberamings ϕ'_n en ϕ''_n as 'n 'nodus' verwys.

Ottoy en De Strycker (2016) het 'n algoritme voorgestel waarvolgens die rigtingberamings van verskeie nodusse gebruik kan word om hierdie onsekerheid tussen ϕ'_n en ϕ''_n te verwyder. Dit kan gedoen word as die klankbron in die **nabye veld** is ten opsigte van die nodusse, maar in die **verre veld** is ten opsigte van die mikrofone in elke nodus. Dit beteken dat die afstand tussen die middelpunte van die verskillende nodusse vergelykbaar is met die afstand tussen die onderskeie nodusse en die klankbron, terwyl die afstand(e) tussen die mikrofone in elke nodus baie klein is met betrekking tot die afstand tussen die betrokke nodus en die klankbron.

Omdat elke ϕ_n in Vergelykings 10.8 en 10.9 een van twee waardes, naamlik ϕ'_n en ϕ''_n , kan aanneem, kan die versameling $\{\phi_1, \phi_2, \dots, \phi_N\}$ op N^2 maniere geïnterpreteer word en is daar N^2 oorgedetermineerde stelsels om op te los, waarvan slegs een die korrekte klankbronposisie lewer. Die algoritme van Ottoy en De Strycker (2016) los elkeen op en soek dan deur die resultate na die korrekte oplossing. Dit kan soos volg opgesom word:

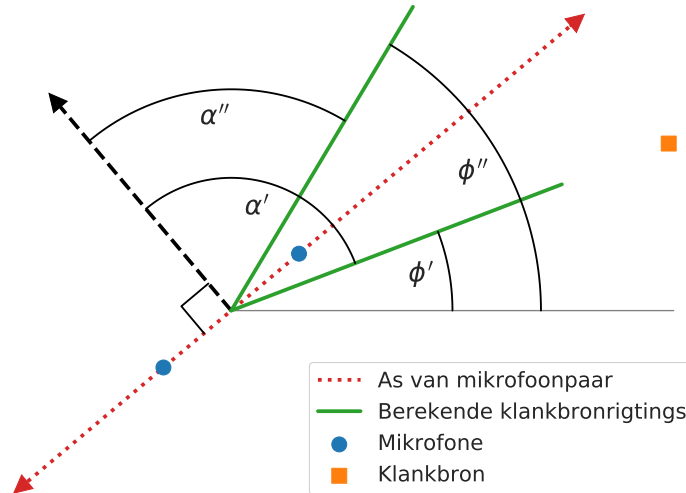


(a) $N_{\text{ruisbisse}} = 4$. Die impak van ruis op die rigtingberaming is hier ongeveer van dieselfde orde as die akkuraatheid van die simulاسie. Om 'n meer gedetailleerde figuur te kry, sou dit nodig wees om die mate van oormonstering in die simulاسie-opstelling te verhoog, wat nie moontlik was met die beskikbare rekenars nie.



(b) $N_{\text{ruisbisse}} = 8$. Hier begin die ruis die beramingsfout domineer. Hierdie resultaat laat blyk dat dit die moeite werd sal wees om in die toekoms die ruisvlakke in die oudiokoppelvlak (§5) en kragbronne (§3) verder te probeer verminder.

Figuur 10.10: Simulasies met verskillende hoeveelhede ongekorreleerde ruis by elke mikrofoon. Dit stel die ruis voor wat in die mikrofone, filters, versyferaar en kragbronne van die opnemer bestaan. Hierdie ruis is nie 'n funksie van die klankbronaafstand nie en veroorsaak dat die **SNR** afneem soos wat die klankbronaafstand toeneem. Soos verwag, word hoeke naby 0° en 180° meer geaffekteer, terwyl die kleinste beramingsfoute naby 90° lê. Die simulاسie waar $N_{\text{ruisbisse}} = 2$ toon geen noemenswaardige verskille nie – die figuur lyk net soos Figuur 10.8 en word dus uitgelaat.



Figuur 10.11: 'n Voorbeeld van dubbelsinnigheid van die klankbronrigting. Die posisies aan die een kant van die lyn waarop die mikrofoonpaar lê, kan nie onderskei word van die posisies aan die ander kant van die lyn nie.

1. Vir elkeen van die N^2 interpretasies van $\{\phi_1, \phi_2, \dots, \phi_N\}$:
 - 1.1. $[\hat{p}_x, \hat{p}_y] \leftarrow$ oplossing van Vergelyking 10.9 met behulp van die kleinste kwadratemetode.
 - 1.2. $\hat{\phi}_n \leftarrow \arctan\left(\frac{\hat{p}_y - q_{ny}}{\hat{p}_x - q_{nx}}\right)$ = die hoek wat die nodus n sou meet indien hierdie die regte interpretasie was.
 - 1.3. $\phi_{\text{fout}} \leftarrow \sum_{n=1}^N |\phi_n - \hat{\phi}_n|$
2. Kies die interpretasie met die kleinste ϕ_{fout} .

Ten minste 3 nodusse (9 interpretasies) is nodig vir hierdie algoritme om te werk – anders is Vergelyking 10.9 nie oorgedetermineer nie en is $\phi_{\text{fout}} = 0$ vir al 4 interpretasies.

10.4 Rigtingberaming met 'n nie-lineêre mikrofoonskikking

Die algoritme van Ottoy en De Strycker (2016) het die skrywer gelei om 'n soortgelyke algoritme uit te dink wat die hoekdubbelsinnigheid hanteer vir 'n nie-lineêre mikrofoonskikking, waar die klankbron in die verreveld is ten opsigte van die mikrofone in die skikking. Nadat 'n ondubbelsinnige rigtingberaming vir elke nie-lineêre skikking bereken is, kan triangulasie (soos beskryf in §10.2) uitgevoer word om die klankbronposisie te bepaal.

10.4.1 Algoritme

Elke mikrofoonpaar in die skikking word beskou as 'n afsonderlike lineêre skikking met 2 elemente. Soos voorheen bied elke mikrofoonpaar, oftewel nodus, twee hoeke, naamlik ϕ_n'' en ϕ_n' , en is daar N^2 moontlike interpretasies van die versameling $\{\phi_1, \phi_2, \dots, \phi_N\}$ waar N die aantal nodusse is.

Vir elke interpretasie van die versameling word daar 'n een-veranderlike von Mises-kerndigtheidsberaming met al die hoeke in die versameling bereken. Die von Mises-waarskynlikheidsverdeling is 'n benadering vir die sirkulêre normaalverdeling en word deur Vergelyking 10.10 gedefinieer, waar I_0 die aangepaste nulde-orde Bessel-funksie is. Die konstante κ bepaal hoe nou die waarskynlikheidsverdeling is – kleiner waardes maak die lob breër.

$$f_{\text{kern}}(\phi|\mu, \kappa) = \frac{\exp(\kappa \cos(\phi - \mu))}{2\pi I_0(\kappa)} \quad (10.10)$$

Die **kerndigtheidsberaming** word dus deur Vergelyking 10.11 gegee.

$$g(\phi|\kappa) = \frac{1}{N} \sum_{n=1}^N f_{\text{kern}}(\phi|\mu = \phi_n, \kappa) \quad (10.11)$$

Onder die aanname dat die afstand na die klankbron groot is in vergelyking met die afstand tussen elke mikrofoon, is al die **mikrofoonpare** ongeveer op dieselfde plek. Dan kan die modus van $g(\phi|\kappa)$ beskou word as die konsensus-rigtingberaming $\hat{\phi}$ van al die **mikrofoonpare**, soos aangedui in Vergelyking 10.12.

$$\hat{\phi} = \arg \max_{\phi} (g(\phi)) \quad (10.12)$$

Vir elke versameling $\{\phi_1, \phi_2, \dots, \phi_N\}$ word $\hat{\phi}$ bereken, en die versameling waar al die hoeke so na as moontlik aan die modus is, word as die korrekte intepretasie beskou. Vergelyking 10.13 toon dus die fout wat geminimeer word.

$$\phi_{\text{fout}} = \sum_{n=1}^N |\phi_n - \hat{\phi}| \quad (10.13)$$

Die finale klankbronrigtingberaming is dus die waarde van $\hat{\phi}$ wat ϕ_{fout} minimeer.

Die algoritme kan soos volg opgesom word:

1. Vir elkeen van die N^2 interpretasies van $\{\phi_1, \phi_2, \dots, \phi_N\}$:
 - 1.1. $g(\phi) \leftarrow \sum_{n=1}^N f_{\text{kern}}(\phi|\mu = \phi_n, \kappa) =$ 'n sirkulêre von Mises-**kerndigtheidsberaming**.
 - 1.2. $\hat{\phi} \leftarrow \arg \max (g(\phi)) =$ die modus van die **kerndigtheidsberaming** (neem aan dat daar net een klankbron is en dat die **kerndigtheidsberaming** dus **eentoppig** is).
 - 1.3. $\phi_{\text{fout}} \leftarrow \sum_{n=1}^N |\phi_n - \hat{\phi}|$
2. Neem aan dat die nodusse almal dieselfde posisie het⁸, en neem die modus van die interpretasie met die kleinste ϕ_{fout} as die klankbronrigting.

10.4.2 Numeriese implementasie

Omdat slegs die posisie van die top van die waarskynlikheidsverdeling van belang is, kan die noemer $2\pi I_0(\kappa)$ in Vergelyking 10.10 en die normalisering $\frac{1}{N}$ in Vergelyking 10.11 geïgnoreer word. Vergelyking 10.14 herdefinieer die **kerndigtheidsberaming** en gee die eerste twee afgeleides daarvan m.b.t. ϕ .

$$g(\phi|\kappa) = \sum_{n=1}^N \exp(\kappa \cos(\phi - \phi_n)) \quad (10.14a)$$

$$g'(\phi|\kappa) = -\kappa \sum_{n=1}^N \sin(\phi - \phi_n) \exp(\kappa \cos(\phi - \phi_n)) \quad (10.14b)$$

$$g''(\phi|\kappa) = \kappa \sum_{n=1}^N \exp(\kappa \cos(\phi - \phi_n)) (\kappa \sin^2(\phi - \phi_n) - \cos(\phi - \phi_n)) \quad (10.14c)$$

Om die top van $g(\phi|\kappa)$ te vind, kan die **NCG** metode (Nocedal en Wright, 2006, p.168) gebruik word, wat van die eerste en tweede afgeleides gebruik maak. Die **Scipy**-biblioteek

⁸ Dit is 'n aanvaarbare aanname, mits die afstand tussen al die mikrofone in die skikking klein is met betrekking tot die verwagte klankbronaafstand. As 'n olifant nader as 20 cm aan die apparaat kom, maak die wiskunde in elk geval nie meer saak nie.

bied 'n implementasie hiervan (Verskeie outeurs, 2016). Omdat hierdie implementasie slegs minimering doen, moet die tekens van Vergelyking 10.14 omgeruil word. Die minimeringsfunksie `scipy.optimize.minimize` word met die volgende argumente⁹ geroep om hierdie optimering uit te voer:

`fun` Die funksie¹⁰ om te minimeer: $-g(\phi|\kappa)$.

`x0` 'n Eerste benadering vir $\hat{\phi}$.

`method` 'Newton-CG'

`jac` Die eerste afgeleide van die funksie wat geminimeer word: $-g'(\phi|\kappa)$.

`hess` Die tweede afgeleide van die funksie wat geminimeer word: $-g''(\phi|\kappa)$.

Vier strategieë om die beginwaarde `x0` te kry, is ondersoek:

- Gebruik 'n ewekansige hoek in $[-180^\circ, 180^\circ]$.
- Kies een van die hoeke in $\{\phi_1, \phi_2, \dots, \phi_N\}$ lukraak.
- Bereken $g(\phi|\kappa)$ vooraf by 'n paar eweredig gespasieerde waardes van ϕ , en gebruik $\arg \max_{\phi} (g(\phi|\kappa))$ as die eerste benadering vir die minimeringsalgoritme.
 - Bereken dit direk, deur waardes van ϕ in g in te stel.
 - Bereken 'n histogram oor ϕ en neem die vouprodukt van die histogram met die von Mises-kern. Dit is 'n amper-eksakte benadering vir $g(\phi|\kappa)$ soos wat die aantal dromme in die histogram meer word¹¹. Dit bied ook geleentheid vir optimering met behulp van die vinnige Fourier-transform (FFT, Fast Fourier Transform): Die vouprodukt is gelyk aan die inverse DFT van die produk van die DFT van die histogram met die DFT van die kern. Die aantal dromme in die histogram word ook as die aantal punte in die DFT gebruik. Omdat die kern dieselfde bly, kan $F_{\text{kern}} = \text{DFT}(f_{\text{kern}})$ vooraf bereken word vir 'n sekere aantal dromme. Die FFTW-biblioteek (Frigo en Johnson, 2005, 2017; Gomersall *et al.*, 2016) en die `fast_histogram`-biblioteek (Robitaille, 2017) word gebruik, omdat dit baie vinniger FFTs en histogramme kan bereken as die Numpy- en SciPy-biblioteke wat in die res van die projek gebruik word.

Die eerste twee oplossings sal slegs werk as die waarskynlikheidsverdeling werklik eentoppig is, maar is die eenvoudigste om te implementeer. Vir 'n meertoppige verdeling sal die vier strategieë nie altyd dieselfde antwoord gee nie. Figuur 10.12 toon die resultate vir 'n versameling $\{\phi_1, \phi_2, \dots, \phi_N\}$ wat so 'n verdeling veroorsaak. Hierdie probleem word egter deur die volgende stap in die algoritme opgelos, omdat slegs die verdeling waar al die hoeke so na as moontlik aan die modus is, gekies sal word. Daardie verdeling behoort wel eentoppig te wees.

Omdat NCG vinniger konvergeer as die beginpunt `x0` naby die finale antwoord $\hat{\phi}$ is, is die eerste twee strategieë nie noodwendig die vinnigste oplossings nie. Die volgende simulatie is uitgevoer om die verwerkingstyd van elke strategie te bepaal:

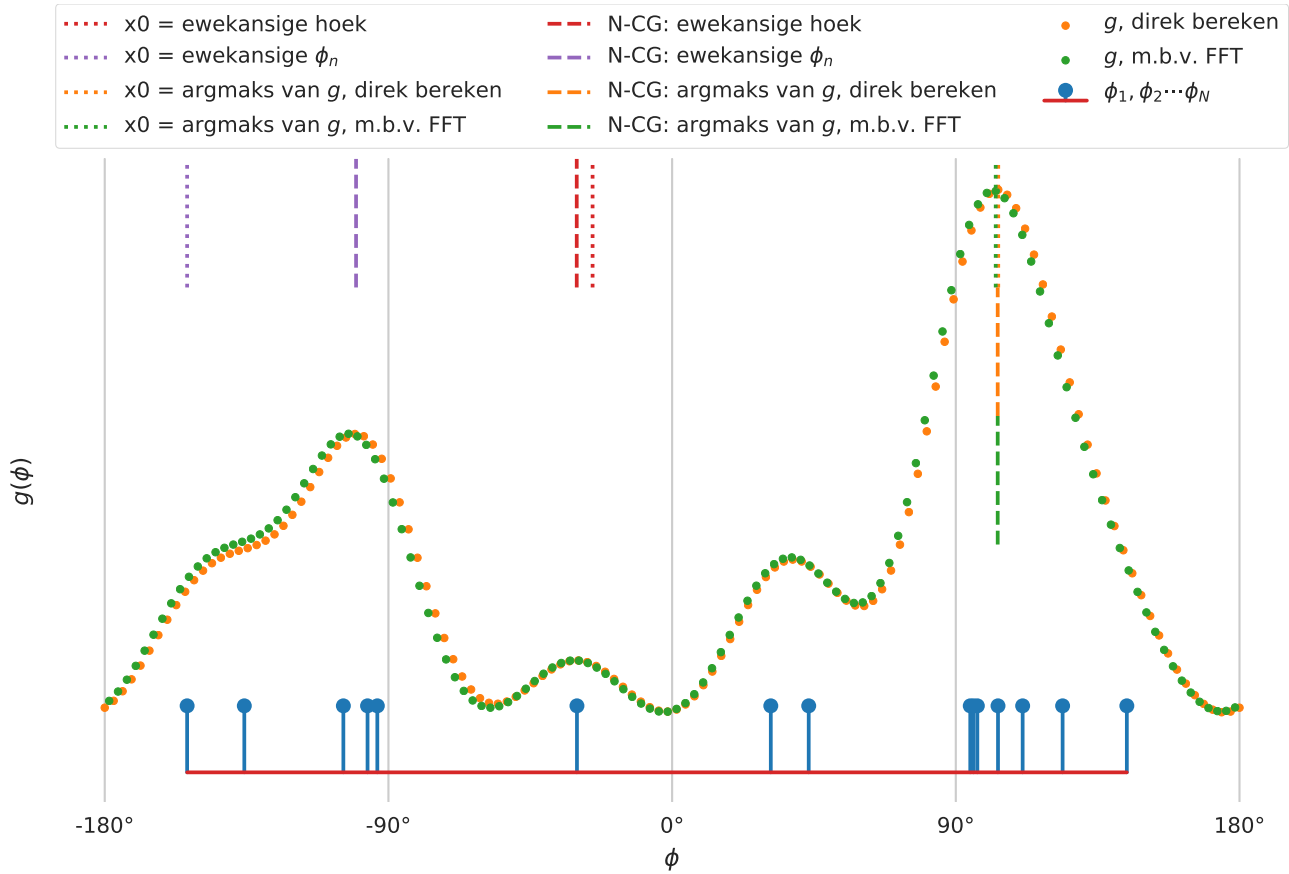
1. Skep 100 000 versamelings van $\{\phi_1, \phi_2, \dots, \phi_N\}$, met $N = 15$. Die waardes van ϕ_n word uit 'n normaalverdeling met 'n standaardafwyking¹² van $22,5^\circ$ en 'n ewekansige gemiddeld getrek en genormaliseer om in $[-180^\circ; 180^\circ]$ te lê.

⁹ Die name van die argumente word deur die SciPy-biblioteek bepaal.

¹⁰ Die wiskundige funksies g , g' en g'' is as Python-funksies geïmplementeer wat vir elke float-insetwaarde ϕ 'n float teruggee.

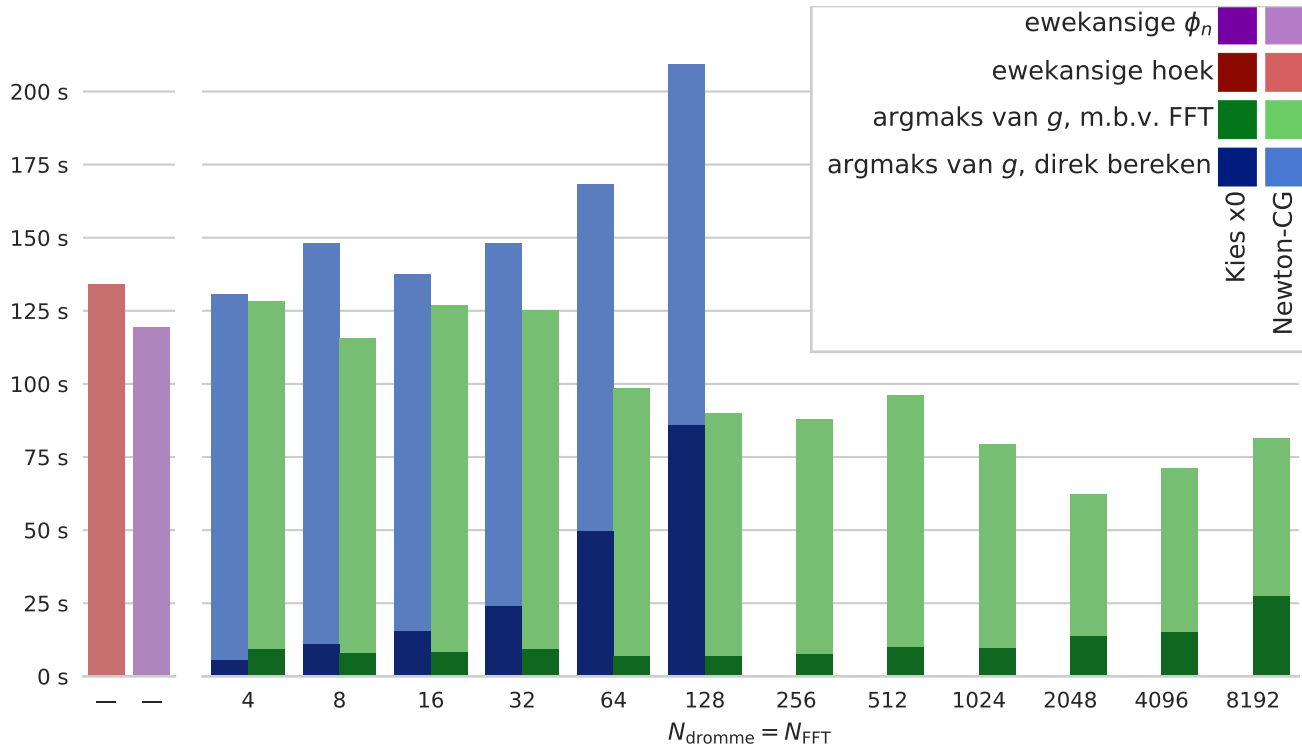
¹¹ Dit werk omdat die kerndigtheidsberaming in Vergelyking 10.14a ook gedefinieer kan word as die vouprodukt van die kern met 'n funksie wat 1 is as $\phi = \phi_n \forall n$ en 0 andersins.

¹² Die standaardafwyking word gekies sodat die waarskynlikheidsverdelings ongeveer lyk soos dié wat later in Figuur 10.14 getoon word.



Figuur 10.12: 'n Voorbeeld van 'n datasetel $\{\phi_1, \phi_2, \dots, \phi_N\}$ wat 'n meertoppige waarskynlikheidsverdeling g veroorsaak. Die vier strategieë vir die bepaling van die posisie van die top van g bied dus verskillende resultate. Hierdie probleem word opgelos deurdat stap 2 van die algoritme in §10.4.1 juis die mees eentoppige verdeling uitsoek.

2. Vir elke versameling:
 - 2.1. $x_0 \leftarrow$ 'n ewekansige hoek in $[-180^\circ, 180^\circ]$.
 - 2.2. Vind die top met behulp van NCG.
3. Vir elke versameling:
 - 3.1. $x_0 \leftarrow$ 'n ewekansige hoek uit die versameling.
 - 3.2. Vind die top met behulp van NCG.
4. By verskeie waardes van $N_{\text{dromme}} = N_{\text{FFT}}$:
 - 4.1. Vir elke versameling:
 - 4.1.1. Bereken $g(\phi|\kappa)$ by N_{dromme} eweredig gespaseerde waardes van $\phi \in [-180^\circ, 180^\circ]$
 - 4.1.2. $x_0 \leftarrow \arg \max_{\phi} (g(\phi|\kappa))$
 - 4.1.3. Vind die top met behulp van NCG.
 - 4.2. Vir elke versameling:
 - 4.2.1. $h \leftarrow$ 'n histogram van die versameling, met N_{dromme} dromme.
 - 4.2.2. $\hat{g}(\phi|\kappa) \leftarrow \text{IDFT}(\text{DFT}(h) \cdot F_{\text{kern}})$
 - 4.2.3. $x_0 \leftarrow \arg \max_{\phi} (\hat{g}(\phi|\kappa))$
 - 4.2.4. Vind die top met behulp van NCG.



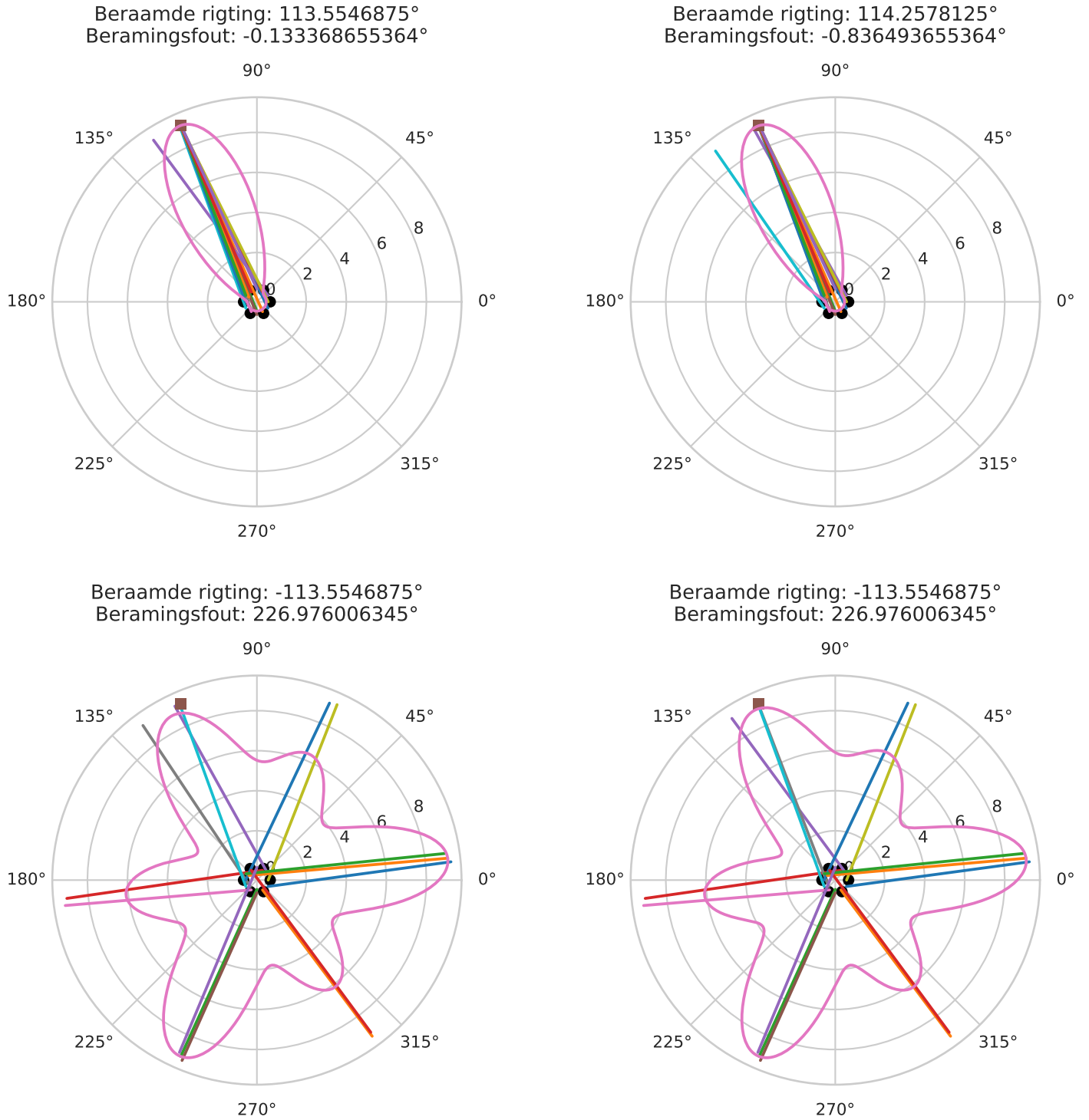
Figuur 10.13: 'n Vergelyking van uitvoertyd van vier verskillende strategieë vir die bepaling van die top van g , vir 100 000 verskillende datastelle. Die donker kleure (onderste deel van elke balk) stel die tyd voor wat dit neem om die beginpunt \mathbf{x}_0 te kies. Die ligter kleure (boonste deel van elke balk) is die tyd wat dit neem vir die **NCG**-algoritme om te konvergeer. Om 'n ewekansige hoek te kies, is so vinnig dat die donker balke nie sigbaar is nie en al die tyd word deur die **NCG**-algoritme in beslag geneem. Die uitvoertyd van die direkte metode neem vinnig toe soos wat die aantal dromme meer word en is nie getoets vir meer as 128 dromme nie. Die **FFT**-benadering van g bied die vinnigste resultate met 2048 dromme, wat met 'n resolusie van $\frac{360^\circ}{2048} \approx 0,176^\circ$ ooreenstem.

Die tyd wat dit neem om elke stap uit te voer, is gemeet en word in Figuur 10.13 opgesom. Die eerste twee metodes, waar ewekansige hoeke gekies word, is vinniger as die metode wat op direkte berekening van g staatmaak. Die **FFT**-benadering van g bied egter die heel vinnigste oplossing. Alhoewel die uitvoering van die **FFTs** langduriger word soos wat N_{FFT} toeneem, word die totale tyd minder, omdat die **NCG** vinniger konvergeer.

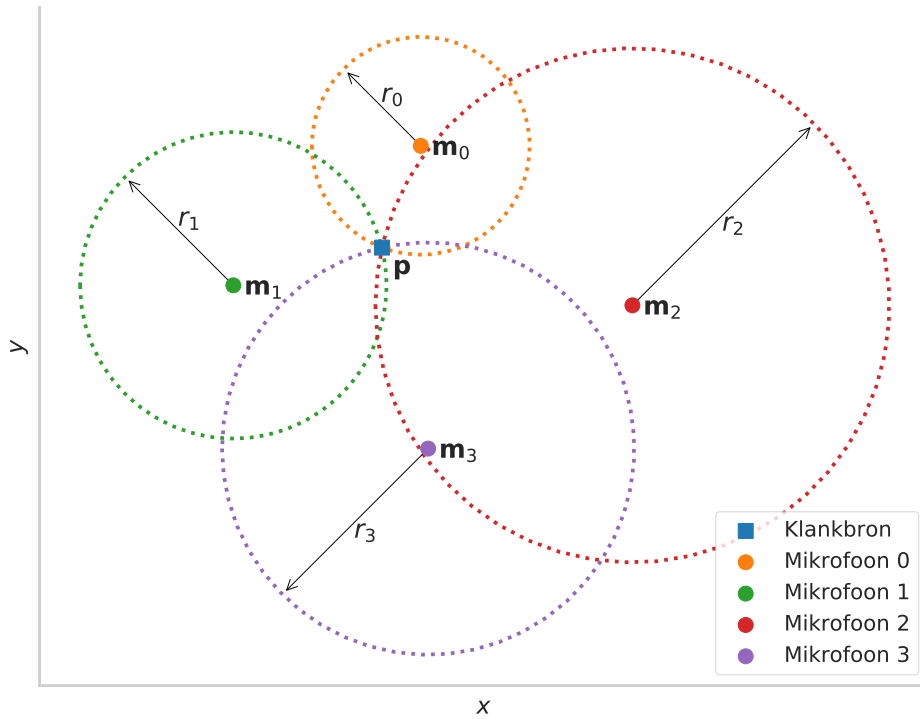
Die bogenoemde simulase is uitgevoer met $\kappa = 10$. 'n Hoë waarde van κ veroorsaak 'n nou, hoë piek in f_{kern} , wat die kans op meertoppigheid in g verhoog, maar die posisie van $\hat{\phi}$ meer bestand maak teen die invloed van uitskieters in $\{\phi_1, \phi_2, \dots, \phi_N\}$. 'n Lae waarde van κ veroorsaak 'n platter waarskynlikheidsverdeling, wat meer waarskynlik eentoppig is, maar die posisie van $\hat{\phi}$ word sterker beïnvloed deur uitskieters. Die keuse van 'n optimale waarde vir κ berus op die werklike toestande waarin die apparaat gebruik gaan word, insluitend die aantal klankbronne in die omgewing en die afstande tussenin. Die optimering van κ staan oor vir toekomstige werk.

10.4.3 Simulasies

Figuur 10.14 toon 'n voorbeeld van die twee beste en twee slegste oplossings uit 'n simulase met ses mikrofone, gepasieer soos in §4.5 beskryf.



Figuur 10.14: 'n Voorbeeld van 'n simulatie met 'n klankbron (bruin vierkant) en ses mikrofone (swart sirkels). Die algoritme in §10.4.1 sorteer die N^2 moontlike interpretasies van $\{\phi_1, \phi_2, \dots, \phi_N\}$ van beste (laagste ϕ_{fout}) na slegste (grootste ϕ_{fout}). Vir elkeen van die interpretasies in stap 1 van die algoritme word die twee beste (bo) en slegste (onder) interpretasies hier getoon. Die korrekte interpretasie van $\{\phi_1, \phi_2, \dots, \phi_N\}$ word dus suksesvol vasgestel. Die modus van die kerndigtheidsberaming (pienk kurwe) van $\{\phi_1, \phi_2, \dots, \phi_N\}$ is die finale beraamde klankbronrigting.



Figuur 10.15: Vier gesinkroniseerde wyd-gepasieerde mikrofone en 'n klankbron in die nabye veld. Drie volledig gedefinieerde sirkels is nodig om die posisie van die klankbron te bepaal. Die absolute afstande (radiusse) is egter onbekend. Vier of meer afstandsverskille (Vergelyking 10.17) word dus benodig om die klankbronposisie te bepaal.

10.5 Nabye-veld-veelsymeting met gesinkroniseerde wyd-gepasieerde mikrofone

'n Alternatiewe benadering vir die opsporing van klankbronne is met **veelsymeting**. Dit is soortgelyk aan die algoritmes wat deur **GPS**-modules gebruik word om hul ligging te bepaal, behalwe dat verskeie sensors gebruik word om een klankbron te vind, eerder as om die posisie van die sensor af te lei van die posisies van die senders.

Bancroft (1985) was die eerste om geslote oplossings vir posisies van **GPS**-ontvangers te bepaal op grond van die aankomstydverskille van seine wat van **GPS**-satelliete afkomstig is. Fang (1990) het 'n soortgelyke oplossing vir 'n Cartesiese vlak ontwikkel. Die probleem kan as 'n stel lineêre vergelykings voorgestel word, waarvoor die afleiding¹³ volg.

10.5.1 Afleiding

Met die klankbron by posisie $\mathbf{p} = [p_x, p_y]$ en N gesinkroniseerde mikrofone by posisies $\mathbf{m}_n = [m_{nx}, m_{ny}]$ waar $n \in \{0, 1, \dots, N-1\}$, word die kwadraat van die afstand vanaf mikrofoon n na die klankbron deur Vergelyking 10.15 gegee. Figuur 10.15 toon 'n scenario waar $N = 4$

$$r_n^2 = (p_x - m_{nx})^2 + (p_y - m_{ny})^2 \quad (10.15)$$

Om die wiskunde te vereenvoudig, word die oorsprong by mikrofoon \mathbf{m}_0 geplaas, sodat $r_0^2 = p_x^2 + p_y^2$.

Die absolute tyd t_n wat dit die klank neem om van die klankbron tot by mikrofoon n te propageer, is nie bekend nie, maar die verskille tussen die aankomstye by elke mikrofoon kan

¹³ 'n Meer toeganklike wiskundige afleiding as die van Bancroft en Fang (soortgelyk aan die een wat in hierdie hoofstuk gegee word), is aanlyn by https://en.wikipedia.org/wiki/Multilateration#Solution_algorithms beskikbaar [2017-12-07].

met behulp van kruiskorrelasies beraam word (sien §10.1). As mikrofoon 0 as tyd-verwysing gebruik word, word die verband tussen die tydverskille τ_n en die absolute aankomstye t_n deur Vergelyking 10.16 gegee.

$$\tau_n = t_n - t_0 \quad (10.16)$$

Die **afstandsverskille** word verkry deur Vergelyking 10.16 met die spoed van klank v te vermenigvuldig:

$$\begin{aligned} r_n - r_0 &= v * \tau_n \\ \therefore r_n &= r_0 + v * \tau_n \end{aligned} \quad (10.17)$$

As die presiese afstande r_n vir alle n bekend was, sou 3 mikrofone genoeg wees om die klankbron op te spoor, omdat dit die unieke snypunt van drie sirkels is. Omdat slegs die **afstandsverskille** $r_n - r_0$ bekend is, is een ekstra mikrofoon nodig om 'n oplossing te kry¹⁴. Meer as 4 mikrofone veroorsaak 'n oorgedetermineerde stelsel, wat met behulp van die **kleinstekwadratemetode** opgelos kan word.

Neem aan dat $\tau_n \neq 0 \forall n$. Gevalle waar $\tau_n = 0$ word in §10.5.3 hanteer. Kwadreer en herrangskik Vergelyking 10.17:

$$\begin{aligned} r_n^2 &= (r_0 + v\tau_n)^2 \\ \therefore 0 &= v\tau_n + 2r_0 + \frac{r_0^2 - r_n^2}{v\tau_n} \end{aligned} \quad (10.18a)$$

$$\text{as } n = 1 : \quad 0 = v\tau_1 + 2r_0 + \frac{r_0^2 - r_1^2}{v\tau_1} \quad (10.18b)$$

Trek Vergelyking 10.18b van Vergelyking 10.18a af:

$$0 = v\tau_n - v\tau_1 + \frac{r_0^2 - r_n^2}{v\tau_n} - \frac{r_0^2 - r_1^2}{v\tau_1} \quad (10.19)$$

Herrangskik Vergelyking 10.15:

$$\begin{aligned} r_n^2 &= p_x^2 + p_y^2 - 2p_x m_{nx} - 2p_y m_{ny} + m_{nx}^2 + m_{ny}^2 \\ &= r_0^2 - 2p_x m_{nx} - 2p_y m_{ny} + m_{nx}^2 + m_{ny}^2 \\ \therefore r_0^2 - r_n^2 &= 2p_x m_{nx} + 2p_y m_{ny} - m_{nx}^2 - m_{ny}^2 \end{aligned} \quad (10.20a)$$

$$\text{as } n = 1 : \quad r_0^2 - r_1^2 = 2p_x m_{1x} + 2p_y m_{1y} - m_{1x}^2 - m_{1y}^2 \quad (10.20b)$$

Stel Vergelykings 10.20a en 10.20b in Vergelyking 10.19 in:

$$\begin{aligned} 0 &= v\tau_n - v\tau_1 + \frac{2p_x m_{nx} + 2p_y m_{ny} - m_{nx}^2 - m_{ny}^2}{v\tau_n} - \frac{2p_x m_{1x} + 2p_y m_{1y} - m_{1x}^2 - m_{1y}^2}{v\tau_1} \\ \text{oftewel } B_n &= C_n p_x + D_n p_y \\ \text{waar } B_n &= -v\tau_n + v\tau_1 - \frac{m_{nx}^2 + m_{ny}^2}{v\tau_n} + \frac{m_{1x}^2 + m_{1y}^2}{v\tau_1} \\ C_n &= \frac{2m_{nx}}{v\tau_n} - \frac{2m_{1x}}{v\tau_1} \\ D_n &= \frac{2m_{ny}}{v\tau_n} - \frac{2m_{1y}}{v\tau_1} \end{aligned} \quad (10.21)$$

¹⁴ Die absolute radiusse van hierdie sirkels is dus onbekend, maar die verskil tussen die radiusse is bekend. Een ekstra sirkel word dus benodig om die stelsel volledig te determineer.

Herskryf Vergelyking 10.21 in matriksvorm vir $n \in \{2, 3, \dots, N-1\}$:

$$\begin{bmatrix} C_2 & D_2 \\ C_3 & D_3 \\ \vdots & \vdots \\ C_{N-1} & D_{N-1} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} B_2 \\ B_3 \\ \vdots \\ B_{N-1} \end{bmatrix} \quad (10.22)$$

oftewel $\mathbf{A}\mathbf{p} = \mathbf{B}$

Vergelyking 10.22 kan met behulp van 'n **kleinstekwadrade**-oplosser soos `scipy.linalg.lstsq` opgelos word, wat $|\mathbf{A}\mathbf{p} - \mathbf{B}|$ minimeer om 'n benadering vir \mathbf{p} te kry.

10.5.2 Hantering van oortollige inligting

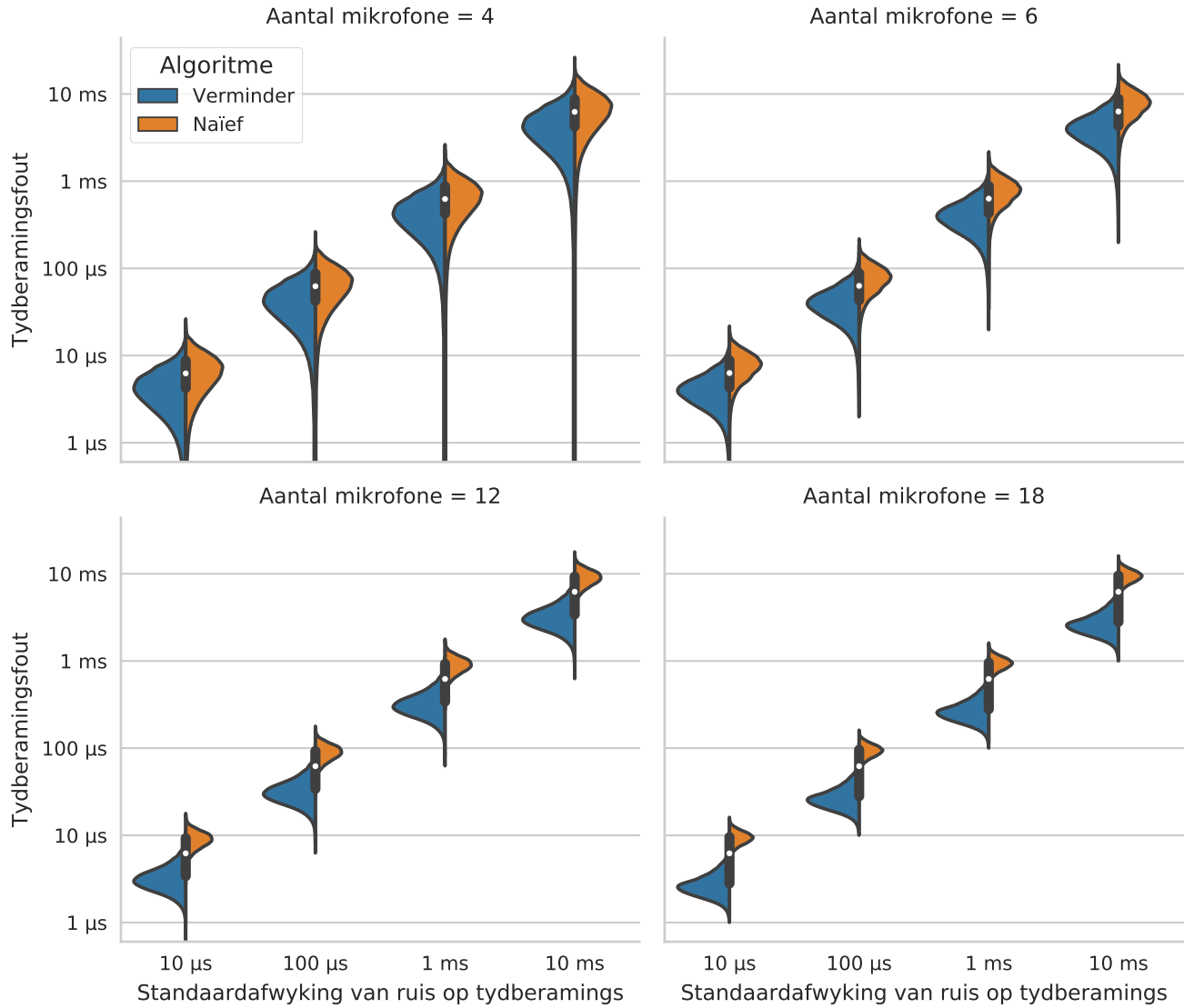
Met N mikrofone is daar $N-1$ relatiewe verdragings τ_n om te beraam, maar daar is $\binom{N}{2}$ mikrofoonpare, en elke paar bied 'n tydvertragsberaming met behulp van kruiskorrelasie soos bespreek in §10.1. As die beraamde tydvertraging tussen mikrofone i en j deur $t_{i,j}$ voorgestel word, kan die verband tussen die onbekendes $\tau_n \forall n \in \{1, 2, \dots, N-1\}$ en die bekendes $t_{i,j} \forall i, j \in \{1, 2, \dots, N-1\}$ waar $i < j$ soos volg voorgestel word:

$$= \begin{bmatrix} 0 & t_{0,1} & t_{0,2} & \cdots & t_{0,N-3} & t_{0,N-2} & t_{0,N-1} \\ & 0 & t_{1,2} & \cdots & t_{1,N-3} & t_{1,N-2} & t_{1,N-1} \\ & & 0 & \cdots & t_{2,N-3} & t_{2,N-2} & t_{2,N-1} \\ & & & \ddots & \vdots & \vdots & \vdots \\ & & & & 0 & t_{N-3,N-2} & t_{N-3,N-1} \\ & & & & & 0 & t_{N-2,N-1} \\ & & & & & & 0 \end{bmatrix} \quad (10.23)$$

$$= \begin{bmatrix} 0 & \tau_1 & \tau_2 & \cdots & \tau_{N-3} & \tau_{N-2} & \tau_{N-1} \\ & 0 & \tau_2 - \tau_1 & \cdots & \tau_{N-3} - \tau_1 & \tau_{N-2} - \tau_1 & \tau_{N-1} - \tau_1 \\ & & 0 & \cdots & \tau_{N-3} - \tau_2 & \tau_{N-2} - \tau_2 & \tau_{N-1} - \tau_2 \\ & & & \ddots & \vdots & \vdots & \vdots \\ & & & & 0 & \tau_{N-2} - \tau_{N-3} & \tau_{N-1} - \tau_{N-3} \\ & & & & & 0 & \tau_{N-1} - \tau_{N-2} \\ & & & & & & 0 \end{bmatrix}$$

Vergelyking 10.23 kan soos volg herskryf word, waar I_n 'n eenheidsmatriks met grootte n voorstel:

1. N mikrofone en 1 klankbron word by lukrake posisies binne 'n 1 km radius van die oorsprong geplaas.
2. Die presiese tye t_n wat dit klank sou neem om van die klankbron na 'n mikrofoon te beweeg, word vir elke mikrofoon bereken.
3. $t_{i,j} \leftarrow t_j - t_i =$ ware tydverskille.



Figuur 10.16: Die vermindering van oortollige tydberamings met die **kleinstekwadratemetode** bied kleiner relatiewe tydberamingsfoute in die teenwoordigheid van beramingsfoute op die aankomstydverskille by die individuele mikrofoonpare.

4. $t_{\text{steur},i,j} \leftarrow$ Gaussiese ruis om foute in die tydberamings te simuleer.
5. $\hat{t}_{i,j} \leftarrow t_{i,j} + t_{\text{steur},i,j} =$ beraamde tydverskille.
6. Vergelyk naïewe metode met bogenoemde metode:

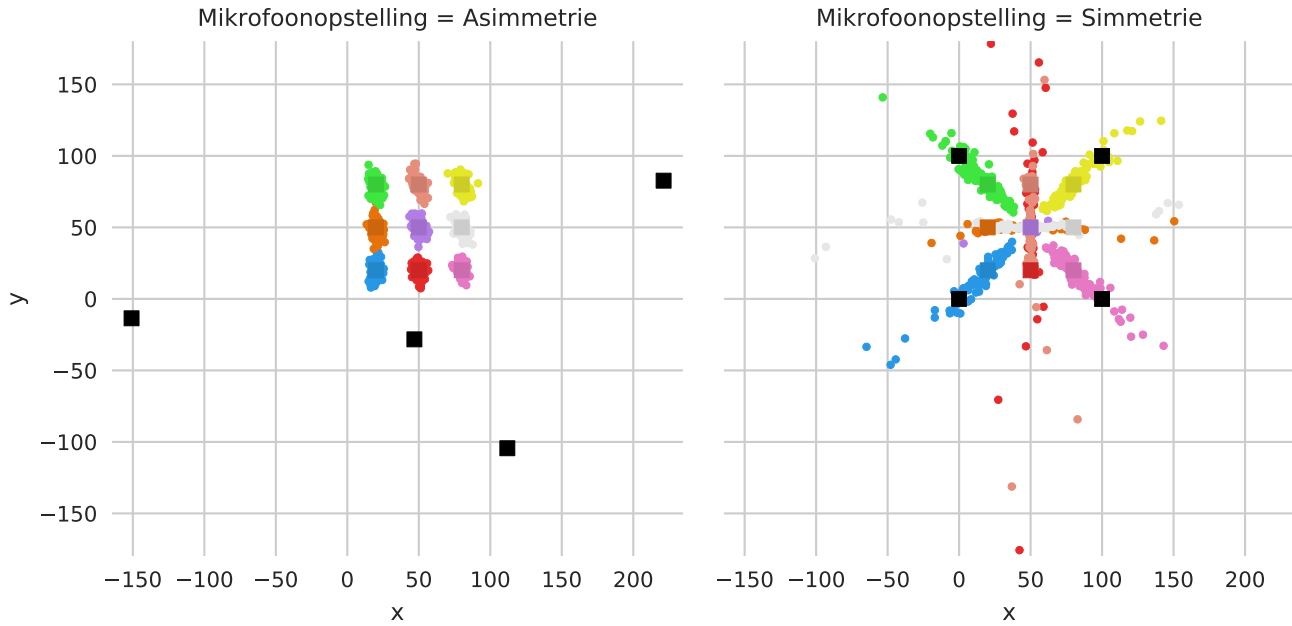
Naïef

- 6.1. Laat $\tau_n = \hat{t}_{0,n}$ volgens die eerste ry in Vergelyking 10.23.
- 6.2. Beramingsfout $= \tau_n - t_{0,n} = t_{\text{ruis},i,j}$

Verminder

- 6.1. Los Vergelyking 10.24 op om τ_n te kry.
- 6.2. Beramingsfout $= \tau_n - t_{0,n}$

Die beramingsfout vir die algoritme wat van al die verskillende beramings gebruik maak, is meer immuun teen ruis. Hierdie resultaat is duidelik sigbaar in Figuur 10.16.



Figuur 10.17: 'n Vergelyking van *veelsymetring* met 'n asimmetriese (links) en simmetriese (regs) mikrofoon-opstelling. Swart vierkante is mikrofone, gekleurde vierkante is klankbronposisies (afsonderlik gesimuleer), en kolletjies van dieselfde kleur is beramings vir die klankbronposisies met behulp van *veelsymetring*. In beide opstellings is daar doelbewus 'n oordrewe hoeveelheid ruis by die tydberamings gevoeg om die verspreiding van posisieberamings uit te lig. In die regter figuur, met 4 mikrofone in 'n simmetriese opstelling, kan die posisies van klankbronne op die simmetrie-asse ($x = 50$ en $y = 50$) nie bepaal word nie, omdat die rang van die matriks wat opgelos word vir alle praktiese doeleindes te laag is. In die linker figuur is die mikrofone lukraak rondgeskuiw, wat maak dat Vergelyking 10.22 wel na wense opgelos kan word.

10.5.3 Hantering van gelyke afstande vanaf klankbron

As een of meer mikrofone ewe ver van die klankbron af is as die verwysingsmikrofoon, is $\tau_n = 0$ vir sommige n . Die afleiding verskil dus van dié in §10.5.1 omdat Vergelykings 10.18a en 10.18b slegs geld vir $\tau_n \neq 0$.

Met $\tau_n = 0$ in Vergelyking 10.17 is $r_n = r_0$. Vervolgens, uit Vergelyking 10.20a:

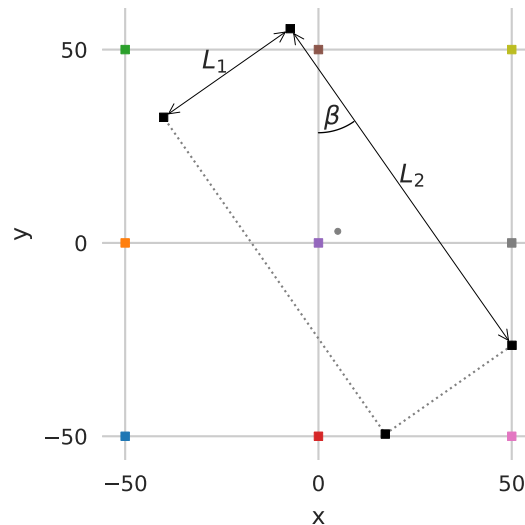
$$m_{nx}^2 + m_{ny}^2 = 2m_{nx}p_x + 2m_{ny}p_y \quad (10.26)$$

Dit kan herskryf word as $B_n = C_n p_x + D_n p_y$ net soos in Vergelyking 10.21 en by die matriks in Vergelyking 10.22 gevoeg word voordat die *kleinstekwadrante*-oplosser uitgevoer word.

As daar meer as een mikrofoonpaar is waarin beide mikrofone ewe ver van die klankbron af is, daal die rang van die matriks in Vergelyking 10.22. As dit slegs naastenby die geval is, word die matriks *sleggeaard*. So 'n matriks kan wel *omgekeer* word, maar die omkering is onderworpe aan groot numeriese foute wat nog groter word soos wat die kolineariteit van die matriksrye (of -kolomme) meer word.

In die teoretiese geval waar 4 mikrofone op die hoeke van 'n vierkant geplaas word, die klankbron presies op een van die vierkant se simmetrie-asse lê, en daar geen ruis in die tydberamings is nie, kan Vergelyking 10.22 nie opgelos word nie. In die werklike geval sal daar ruis op die tydberamings wees, en die oplossing vir die klankbronposisie sal by 'n lukrake posisie langs daardie simmetrie-as lê. Hierdie situasie is gesimuleer en word in Figuur 10.17 uitgebeeld. 'n Wiskundige bewys is in Bylae B.1 beskikbaar.

Die oplossing is om een ekstra mikrofoon by te voeg om die simmetrie te breek en sodoende die rang van die matriks te verhoog. Dit lei tot meer akkurate oplossings vir die klankbronposisies.



Figuur 10.18: 'n Opstelling met 4 mikrofone om te toets watter spasiërings geskik is vir **veelsymetring**. Die swart vierkante stel mikrofone voor, en die gekleurde vierkante is die verskillende posisies waarby die klankbron gesimuleer word.

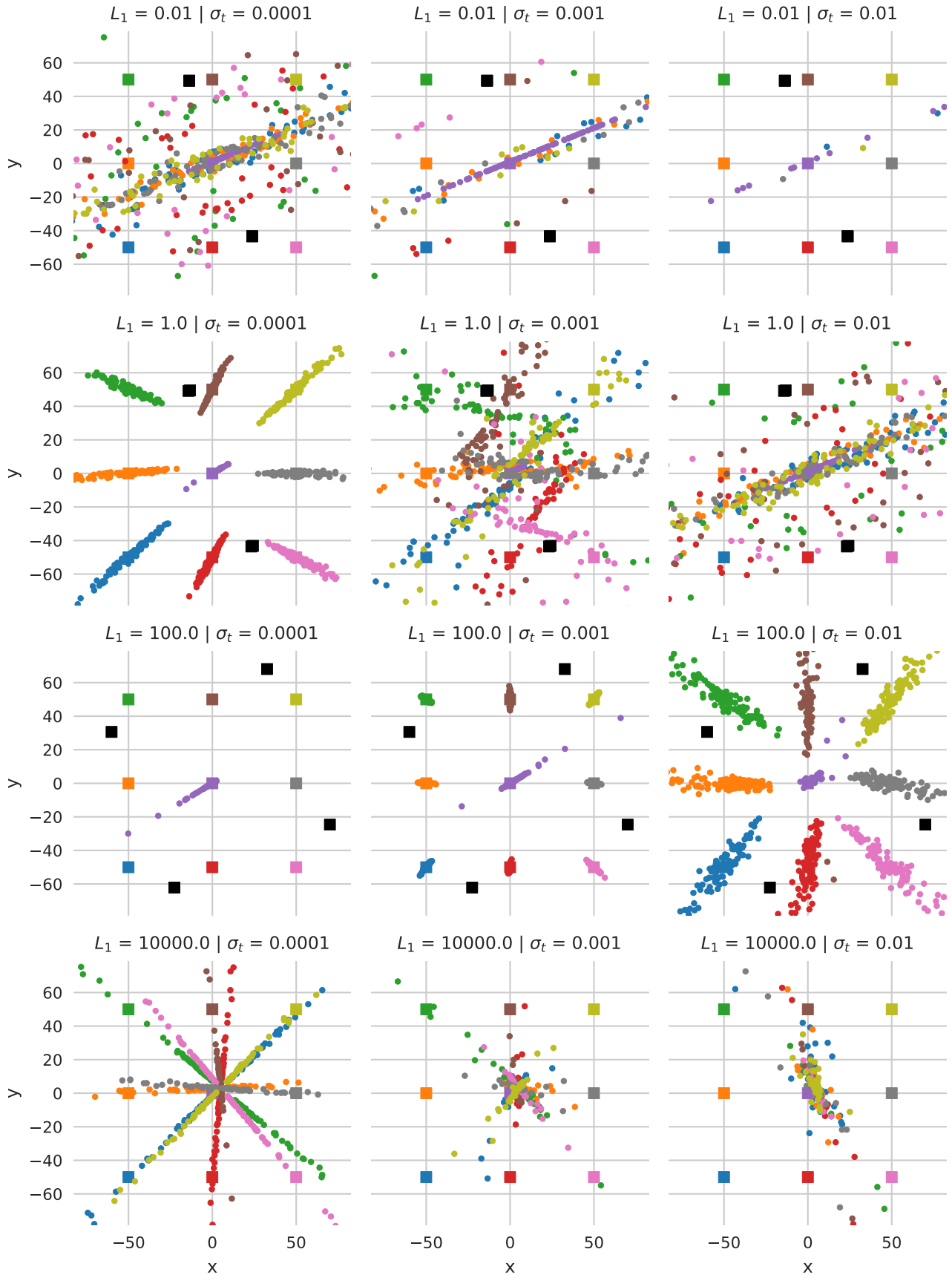
As die klankbron presies op beide simmetrie-asse van die vierkant lê, kan p_x en p_y albei onmiddellik opgelos word deur twee mikrofone se posisies in Vergelyking 10.26 in te stel. As die klankbron egter naby die simmetrie-asse lê, maar nie presies daarop nie, word 'n onbetroubare oplossing vir Vergelyking 10.22 verkry omdat die matriks **sleggeaard** word. 'n Wiskundige bewys is in Bylae B.2 beskikbaar.

10.5.4 Geskikte spasiëring van mikrofone

Dit blyk dat die relatiewe afstande tussen mikrofone 'n groot impak op die klankbronposisieberamings het. 'n Simulasie is met 4 mikrofone uitgevoer om te bepaal wat die impak van nou- en wyd-gespasieerde mikrofone in situasies met verskillende ruisvlakke en verskillende klankbronposisies is. Figuur 10.18 toon die simulasie-opstelling. Die mikrofone lê op die punte van 'n reghoek. L_2 is by 100 m gehou, terwyl L_1 van 1 cm na 10 km gevarieer word. Die middelpunt van die reghoek word by (5 m; 3 m) geplaas en die reghoek word deur $\beta = 22^\circ$ gedraai sodat die mikrofone nie simmetries is m.b.t. die klankbronne nie.

Die simulasie word uitgevoer met verskillende vlakke van ruis (van $\sigma_t = 100 \mu\text{s}$ tot $\sigma_t = 10 \text{ ms}$) op die relatiewe tydberamings τ_n . Figuur 10.19 toon die resultate. Dit is duidelik dat die beste posisieberamings gemaak word as die afstande tussen die mikrofone ongeveer ewe groot is. Wanneer twee of meer mikrofone baie naby aan mekaar is met betrekking tot die afstande tussen die ander mikrofone, ontstaan kolineariteit in die matriksvergelyking 10.22, wat lei tot onakkurate posisieberamings. Hierdie posisieberamings neig dan om op 'n lyn te lê wat loodreg op die lyn tussen die groepies nou-gespasieerde mikrofone is. 'n Wiskundige bewys van hoekom dit gebeur is in Bylae B.3 beskikbaar.

Dit is dus duidelik dat vier of meer wyd-gespasieerde mikrofone benodig word om van **veelsymetring** gebruik te maak. Mikrofone naby mekaar, soos dié van die apparaat wat in hierdie projek ontwerp is, is dus geskik vir rigtingberaming, maar nie veelsymetring nie. Vier of meer afsonderlike, **GPS**-gesinkroniseerde klankopnemers kan egter maklik ingespan word om hierdie probleem op te los. Ongelukkig is daar ten tyde van skrywe slegs 3 sulke eenhede gebou, waarvan een tydens die praktiese toetse as 'n **GPS**-eenheid gebruik is om die ware ligging van die bewegende klankbron vas te stel.



Figuur 10.19: Simulasieresultate met vier mikrofone en verskeie spasiërings. Elke kleur stel 'n aparte klankbronposisie voor. Klankbronne is nie gelyktydig gesimuleer nie, maar afsonderlik. Die gekleurde vierkante is klankbronposisies, en die kolletjies van ooreenstemmende kleure is die posisieberamings. Die swart vierkante stel mikrofone voor. In die eerste twee rye is die mikrofone te na aan mekaar om hulle visueel van mekaar te onderskei. In die laaste ry ($L_1 = 10$ km) is die mikrofone ver buite die omvang van die figuur.

10.6 Toetse met eie klankopnemers

Om te bewys dat beide die hardware-ontwerp en die algoritmes wat in hierdie hoofstuk bespreek is 'n vatbare oplossing vir olifantopsporing is, is praktiese toetse uitgevoer deur ruis en olifantdreungeleude deur 'n luidspreker te speel. Die apparaat moet dus nog in die toekoms op regte olifante getoets word.

10.6.1 Die inlees van opgeneemde data

Die formaat van die data wat die opnemer op die SD-kaart berg, is reeds in §7.5 en Tabel 7.4 beskryf. Elke monster neem 32 B in beslag. Om dit in 'n Python-program in te lees word 'n pasgemaakte Numpy-datatype gedefinieer:

```
monster_tipe = np.dtype(
    [
        ('kanale', np.int32, 6),
        ('ranggetal', np.uint32, 1),
        ('tydstempel', np.uint32, 1)
    ]
)
```

Daarna kan die binêre lêer direk in 'n Numpy-skikking ingelees word:

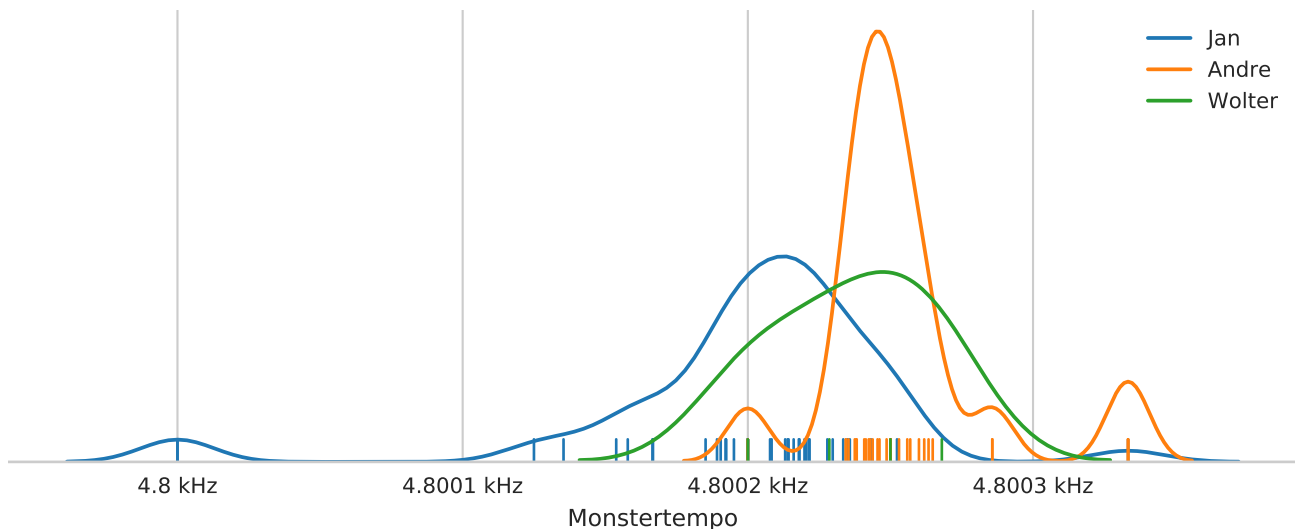
```
data = np.memmap(
    filename=naam_van_lêer,
    dtype=monster_tipe,
    mode='c',
    shape=(aantal_monsters,)
)
```

Die program wat monsters haal en na die SD-kaart skryf is in §7.7 bespreek. As die GPS-sein sou wegval, of die GPS-module stel eers die tyd vas nádat die opname al begin het, sal daar spronge in die tydstempels voorkom. Dit moet met die hand gekorrigeer word. Voorbeelde van hoe dit met toetsopnames gedoen is, is op DVDs E.2 en E.3 in Bylae E beskikbaar.

Die ranggetalle in elke lêer begin by 0, maar dit is onwaarskynlik dat die opname presies op 'n sekondewisseling begin het. In 'n tipiese sekonde loop die ranggetalle van 0 tot 4799¹⁵. In die eerste sekonde loop die ranggetalle egter van 0 tot $N < 4799$, wat 'n skynbare tydsprong by die eerste sekondewisseling veroorsaak. Dit word reggemaak deur die ranggetalle in die eerste sekonde van elke lêer wat ingelees word te herskryf sodat dit van $4799 - N$ tot 4799 loop.

```
# Vind die aanvang van die tweede sekonde in die lêer
i_sekonde2 = next(
    (i for i, t in enumerate(data['tydstempel']) if t != data['tydstempel'][0])
)
if data['ranggetal'][i_sekonde2] != 0:
    raise ValueError("Ranggetal is nie 0 na die sekondewisseling nie. "
                    "Waarskynlik geen GPS-sein. "
                    "Dit moet met die hand reggemaak word...")
# Herskryf ranggetalle
data['ranggetal'][:i_sekonde2] = range(4800 - i_sekonde2, 4800)
```

¹⁵ Omdat die versyferaar se kristal nie met die GPS-klok gesinkroniseer is nie, het sommige sekondes 4801 monsters in plaas van 4800, en loop die ranggetalle van 0 tot 4800.



Figuur 10.20: Verdeling van berekende monstertempos, per apparaat, vir al die toetsopnames wat tydens hierdie projek gemaak is. Die drie klankopnemers wat in hierdie projek gebou is, het die liefderike name ‘Jan’, ‘Andre’ en ‘Wolter’ gekry omdat dit makliker is om te onthou en uit te spreek as ‘Opnemer 1’, ‘Opnemer 2’ en ‘Opnemer 3’. Die toetsdata is op DVDs E.2 en E.3 in Bylae E beskikbaar.

Tot dusver word daar met die aanname gewerk dat die nominale monstertempo van 4800 Hz korrek is. Die ware monstertempo word van die tydstempels en ranggetalle afgelei.

```
eerste = data[0]
laaste = data[-1]
n_monsters = len(data)

t_verskil = laaste['tydstempel'] - eerste['tydstempel']
if t_verskil == 0:
    raise ValueError("Kan nie monstertempo bereken nie, "
                    "want die tydstempel verander nie.")
elif t_verskil < 0:
    raise ValueError("Die volgorde van tydstempels is verkeerd.")

ware_monster_tempo = (
    n_monsters - laaste['ranggetal'] + eerste['ranggetal'] - 1
) / float(t_verskil)
```

Daarna kan 'n presiese tydstop vir elke monster bereken word.

```
presiese_tydstempels = data['tydstempel'] + (
    data['ranggetal'] / ware_monster_tempo
)
```

Die volledige Python-kodes vir die bogenoemde is in die grootvoet Python-biblioteek op DVD E.1 in Bylae E beskikbaar. Die waarskynlikheidsverdeling van ware monstertempos uit al die opnames wat vir hierdie projek gemaak is (nadat GPS-data gekorrigeer is en foutiewe opnames verwyder is), word in Figuur 10.20 getoon.

Omskakeling na WAV-formaat is met die nominale monstertempo gedoen. Dit word slegs gebruik om die opnames maklik te kan beluister nadat toetse gedoen is. Die volledige kodes hiervoor is op DVD E.1 in Bylae E beskikbaar.

Vir die beraming van klankbronposisies moet die ware berekende monstertempos gebruik word. Die monstertempos is nie heeltemal konstant oor tyd nie en verskil ook van een opnemer

Toets	Koördinate van opnemers		Afstand	Klankbron	Figure	
	Jan	Andre			Roete	Resultate
A	-34.227923, 18.412747	-34.227970, 18.412840	10,03 m	Ruis	10.22(a)	10.23
B	-34.227923, 18.412747	-34.227970, 18.412840	10,03 m	Olifantgeluide	10.22(b)	10.24
C	-34.227228, 18.412587	-34.228258, 18.413602	147,69 m	Ruis	10.22(c)	10.25
D	-34.227228, 18.412587	-34.228258, 18.413602	147,69 m	Olifantgeluide	10.22(d)	10.26

Tabel 10.2: Die 4 verskillende opstellings van die toetse wat in 'n stil veld in die Kaapse Skiereiland uitgevoer is. Vir elke toets is beide rigtingberaming en veelsymeting uitgevoer.

na die volgende. Om data byvoorbeeld van 4800,3 Hz na 4800 Hz te hermonster neem baie verwerkertyd. Die ekstra filter-stappe veroorsaak ook onnodige vervorming van die data. Dit is dus beter om hermontering heeltemal te vermy. Om dit te doen, word slim **venstering** toegepas wat die opnames sinkroniseer rondom 'n sekere tydstop, en dan vir elke venster aanneem dat die monstertempos wel konstant is oor tyd en dieselfde is vir elke opnemer. Daarna word die vensters kortgeknip om almal dieselfde lengte te wees. As opnames van twee verskillende klankopnemers met monstertempos van 4800 Hz en 4800,2 Hz op hierdie manier gesinkroniseer word, en vensters van 1 s word gebruik, sal een monster by elke vyfde venster weggegooi word. Die tweede opname word dus teen ongeveer 99,9958 % van die werklike spoed gebruik. Dit veroorsaak slegs 'n klein vervorming van die frekwensiedomein: 500 Hz word byvoorbeeld 499,979 Hz.

10.6.2 Vergelyking van geraas in toetsomgewings

Die omgewingsgeraas het 'n groot impak op die beraming van klankbronposisies. Op 22 Desember 2017 is die rigtingberamingsalgoritme getoets op een van die rugbyvelde naby Coetzenburg, Stellenbosch¹⁶. Daardie tyd van die jaar is daar min mense in Stellenbosch. Dieselfde toets is op 6 Februarie 2018 herhaal, kort na die aanvang van die Universiteit se eerste semester, en baie swakker resultate is verkry.

Dit het gelei tot 'n ondersoek na verskillende toetsterreine om 'n plek te vind wat so stil as moontlik is. By elke plek is 'n paar sekondes 'stilte' opgeneem. 'n **Korttermyn Fourier-transform (STFT, Short Time Fourier Transform)** met 'n vensterlengte van 1 s, 'n treelengte van 0,1 s en 'n Blackmann-apodisasie is gebruik om spektra te verkry. Die gemiddeld van die groottespektrum oor alle vensters en oor al 6 kanale van die opnemer is geneem en op 'n desibelskaal gestip, waar 0 dB met die maksimumwaarde van die versyferaar ooreenstem (dBFS). Figuur 10.21 toon die resultate.

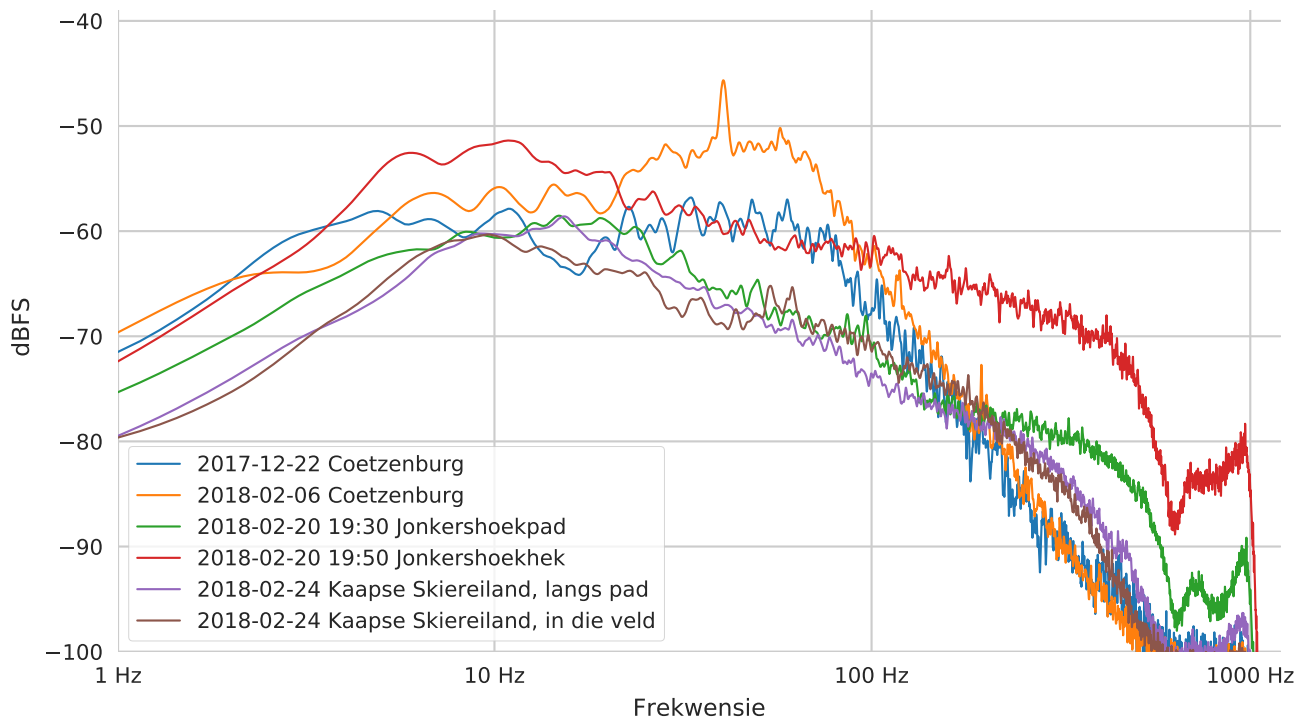
10.6.3 Toetse met twee opnemers en 'n luidspreker in 'n stil omgewing

Daar is besluit om die finale toetse op die stuk veld naby Plateauweg in die Kaapse Skiereiland te doen. Twee klankopnemers is by verskillende posisies in die veld geplaas. Die derde¹⁷ apparaat is as **GPS**-eenheid gebruik om die werklike ligging van die klankbron te bepaal. Die skrywer het dus met 'n battery-aangedrewe luidspreker en **GPS**-eenheid in die veld rondgeloop terwyl die opnemers data versamel.

Tabel 10.2 beskryf die opstelling van die verskillende toetse wat gedoen is. Die afstande tussen die opnemers is bereken deur eers die koördinate met behulp van die **Universele Transversale Mercatorprojeksie (UTM)** na 'n Cartesiese vlak te projekteer en dan Pythagoras se formule toe te pas.

¹⁶ Koördinate: -33,940 495, 18,874 812.

¹⁷ Ongelukkig is die prototipe wat in hierdie projek ontwerp is, net drie keer gebou. Vir toekomstige werk sal dit nuttig wees om meer van hierdie eenhede te bou om te bepaal hoe die akkuraatheid van die stelsel as 'n geheel afhang van die aantal opnemers.



Figuur 10.21: Groottespektra van omgewingsgeraas by verskeie toetsterreine. Stellenbosch is ongeveer 10 dB stiller in Desember (blou) as in Februarie (oranje). Saans langs die Jonkershoekpad (groen) is baie stiller as naby die dorp (blou). Later die aand, naby die ingang na die Jonkershoekreservaat het die wind baie sterk begin waai, wat meer geraas by hoë-frekwensies veroorsaak (rooi). Die heel stilste omgewing wat ondersoek is, is 'n verlate deel van die Kaapse Skiereiland naby Plateauweg, met koördinate $-34,227\ 810, 18,413\ 098$ (pers en bruin).

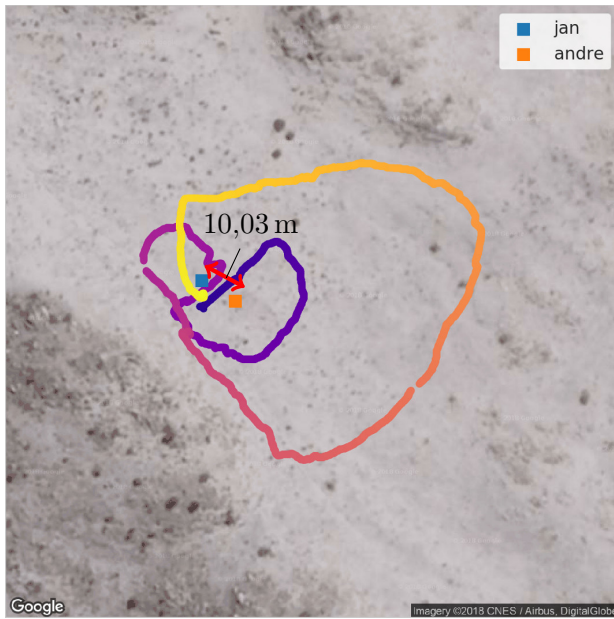
Figuur 10.22(a) toon die roetes wat by elk van die 4 toetse met die luidspreker geloop is. Die veranderende kleur van die lyn wat die roete aandui, dui die verloop van tyd aan. Die vierkante dui die posisies van die opnemers aan.

Vir toets A is die klankopnemers ongeveer 10 m van mekaar af geplaas, en ruis is deur die luidspreker gespeel. Figuur 10.23(a) toon die ware en beraamde klankbronrigtings as 'n funksie van tyd, asook die verskil tussen die ware en beraamde rigtings. Bó die rigtingberamings word 'n spektrogram van een van die mikrofoonseine geteken. Al drie dele is op 'n gemeenskaplike tyd-as gestip. Figuur 10.23(b) toon 'n *kerndigtheidsberaming* van die rigtingfoute vir elke klankopnemer. Soortgelyke figure word vir al 4 toetse gegee (10.23 – 10.26).

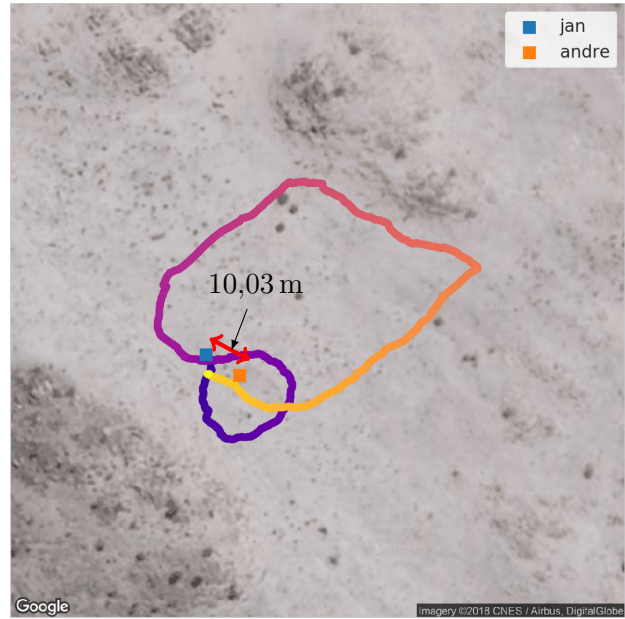
Iets het plaasgevind wat foute in dele van toetse A en B veroorsaak het, en toets C heeltemal ongeldig maak. By 10:11:28 in Figuur 10.23(a) stem die foute ooreen met 'n lae-frekwensie steursein van onbekende oorsprong. By 10:13:58 is daar weer 'n soortgelyke steurnis, maar die beramingsfoute begin al 40 s vroeër en vereis dus 'n ander verklaring. Vermoedelik gebeur dit wanneer die luidspreker ver van die opnemers af is en nie in die rigting van die opnemers wys nie¹⁸. Luidsprekers (en mikrofone) is geneig om meer alsydig te wees by lae frekwensies, en meer gerig by hoë frekwensies. As die luidspreker dus nie in die rigting van die opnemers wys nie is 'n verswakking in die hoër frekwensies te wagte. So 'n verswakking is op die spektrogram sigbaar en stem goed met die tyd van die beramingsfoute ooreen. Verder toon die rigtingberamings (kolletjies) 'n baie goeie korrelasie met die ware rigtings (lyne) wat met behulp van *GPS* gemeet is. Daar is ook 'n onbekende steursein met 'n wisselende fundamentele frekwensie naby 64 Hz wat skynbaar nie die resultate beïnvloed nie.

Vir toets B is die klankopnemers in dieselfde posisie gehou as in toets A, maar olifantdreun- geluide is deur die luidsprekers gespeel. Die geluide is uit die datastel van *Zeppelzauer et al.*

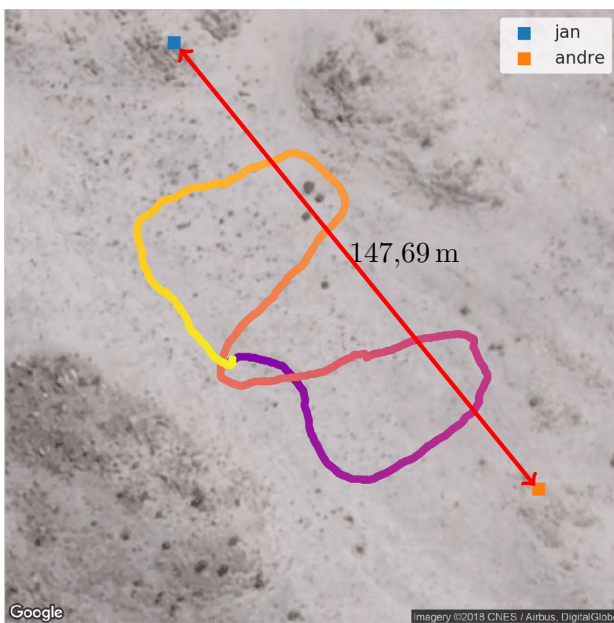
¹⁸ Die rigting waarin die luidspreker wys is ongelukkig nie in ag geneem toe die toetse uitgevoer is nie.



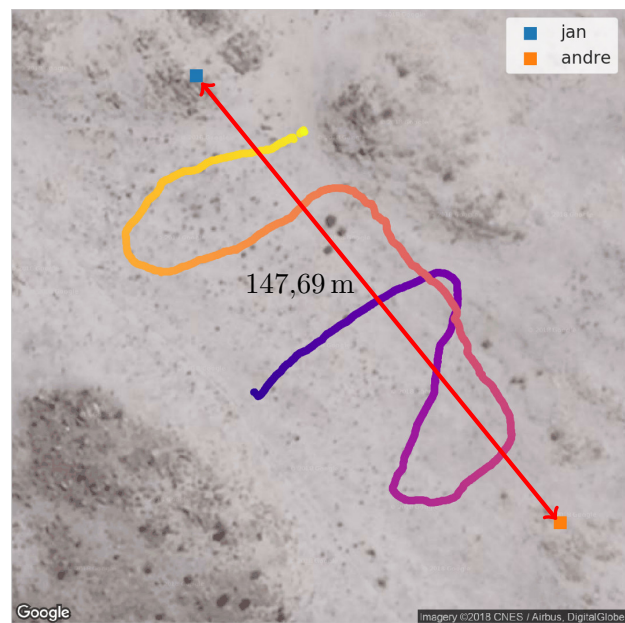
(a) Toets A.



(b) Toets B.

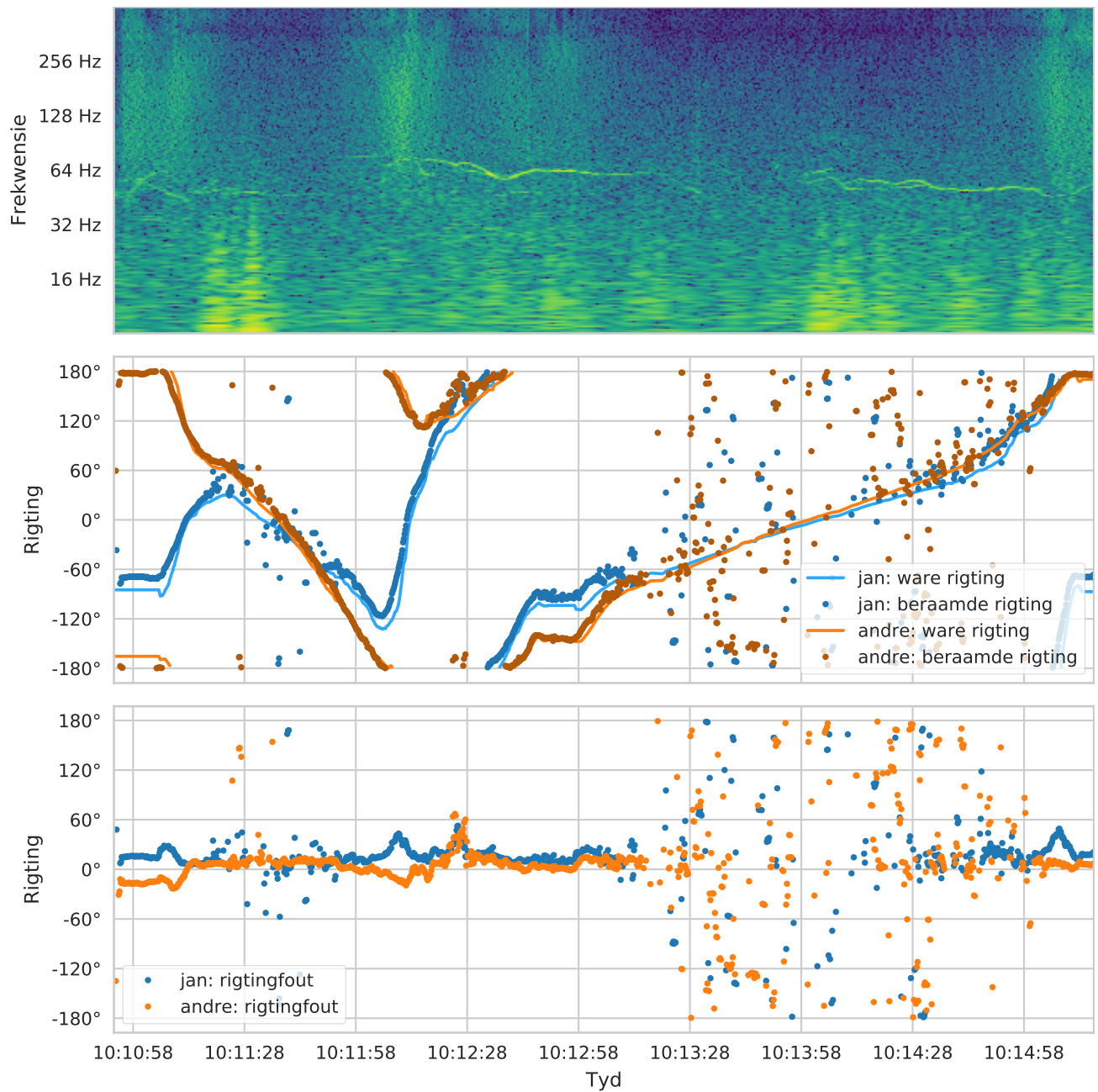


(c) Toets C.

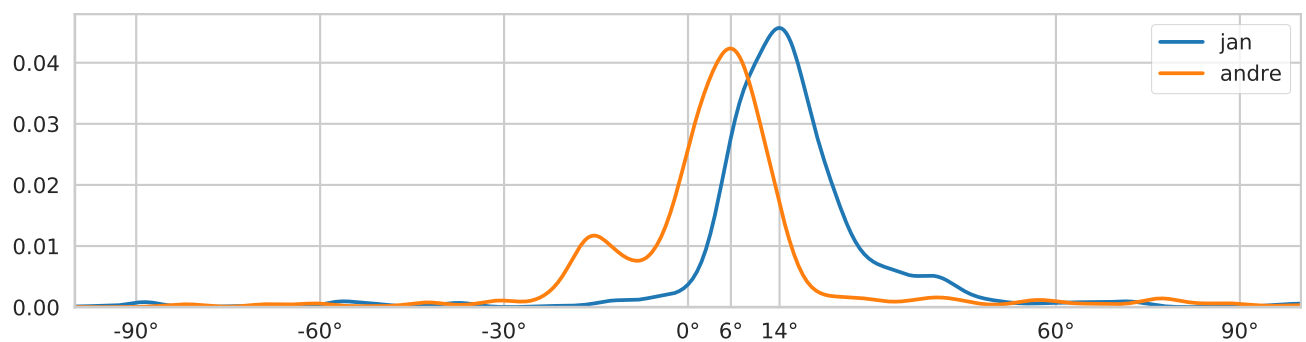


(d) Toets D.

Figuur 10.22: Die roetes van die klankbron vir die onderskeie toetse. Die kleurverandering van pers na geel stel tyd tyd voor. Die agtergrond van hierdie prentjie is deur Google Maps verskaf.



(a) Van bo na onder: Spektrogram van een van die mikrofoonseine tydens hierdie toets; Rigtingberamings; Rigtingberamingsfoute. Die tyd-as word gedeel.



(b) Kerndigtheidsberaming van rigtingfoute.

Figuur 10.23: Rigtingberamingsfoute per opnemer vir toets A, met naby-gespasieerde apparate en 'n ruisbronsin.

(2013) afkomstig. Vyf dreungeluide van ongeveer 2–4 s elk is herhaaldelik deur die luidspreker gespeel, met minimale pouses tussenin. Dit kan gesien word uit die periodiese gedrag van die spektrogram in Figuur 10.24(a). Van ongeveer 10:42:30 tot 10:43:19 is daar 'n lae-frekwensie breë-band steursein wat goed ooreenstem met die skielike rigtingberamingsfoute. Daar is ook 'n harmoniese steursein met sterk komponente by ongeveer 40 Hz en 80 Hz van 10:42:19 tot die einde. Die luidspreker is by 10:43:49 afgeskakel – dit is op beide die spektrogram en die rigtingberamings sigbaar.

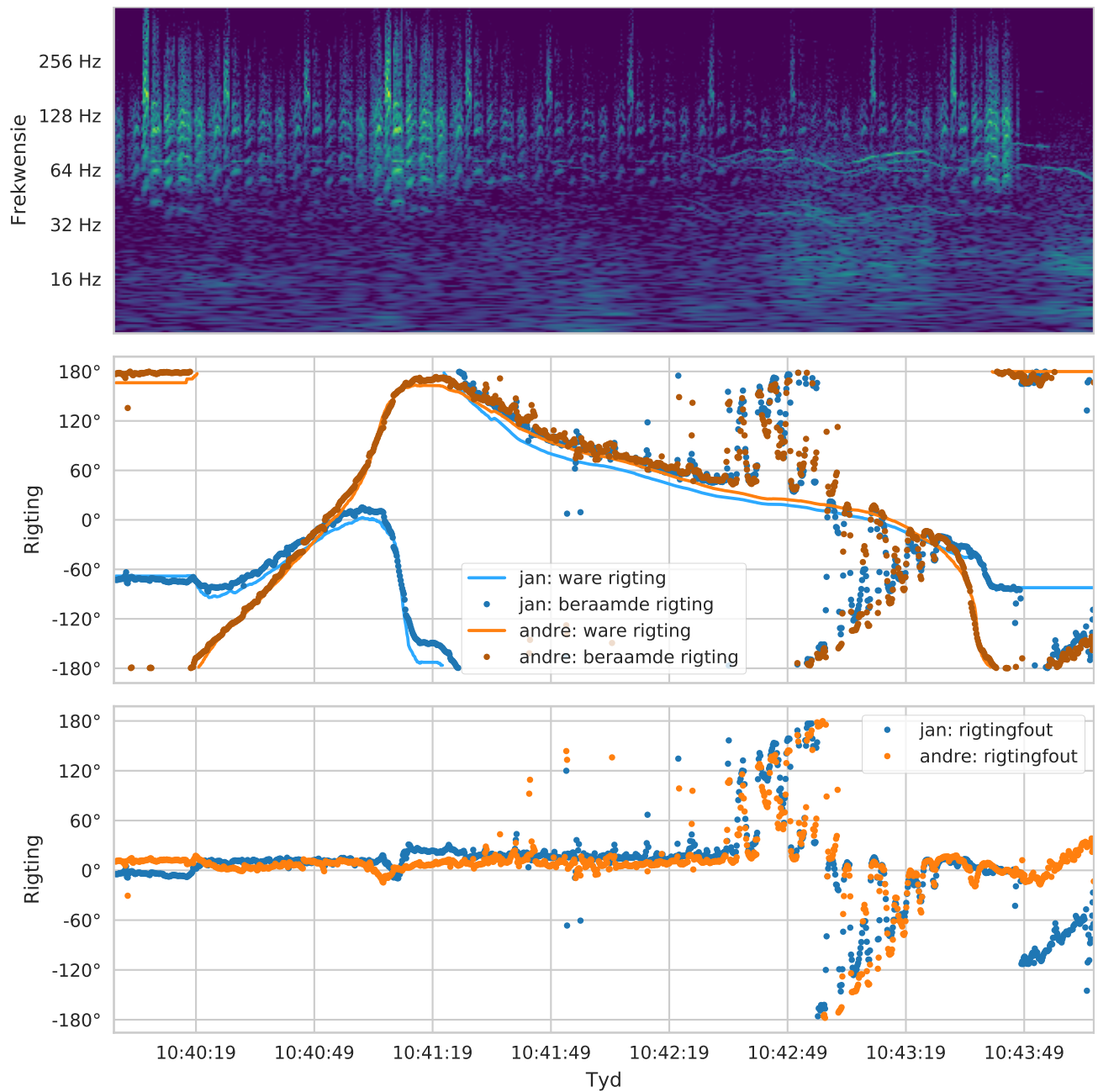
Vir toetse C en D is die opnemers ver uitmekaar geskuif, met ongeveer 148 m tussenin. By toets C is ruis deur die luidspreker gespeel, net soos by toets A. Die toets is egter ongeldig omdat 'n groot breë-band steursein met baie lae-frekwensie energie die rigtingberamings onmoontlik gemaak het.

Toets D is direk na toets C uitgevoer sonder om enigiets aan die opstelling te verander of die opname te onderbreek. Die ruisbron is afgeskakel, en dieselfde olifantgeluide wat in toets B gebruik is, is deur die luidspreker gespeel. Die verwagting was dat hierdie toets die slegste sou vaar omdat a) die klankbron verder van die opnemers af is as in toetse A en B, en b) die klankbron 'n nouer bandwydte het as die ruis wat in toetse A en C gebruik is¹⁹. Die resultate in Figuur 10.26 lyk egter baie belowend. In die toekoms sou 'n tegniek soos 'n Kalman-filter of 'Probability Hypothesis Density' (PHD) filter gebruik kon word om uitskieters te verwyder en die data tot 'n realistiese beraming van die roete saam te voeg.

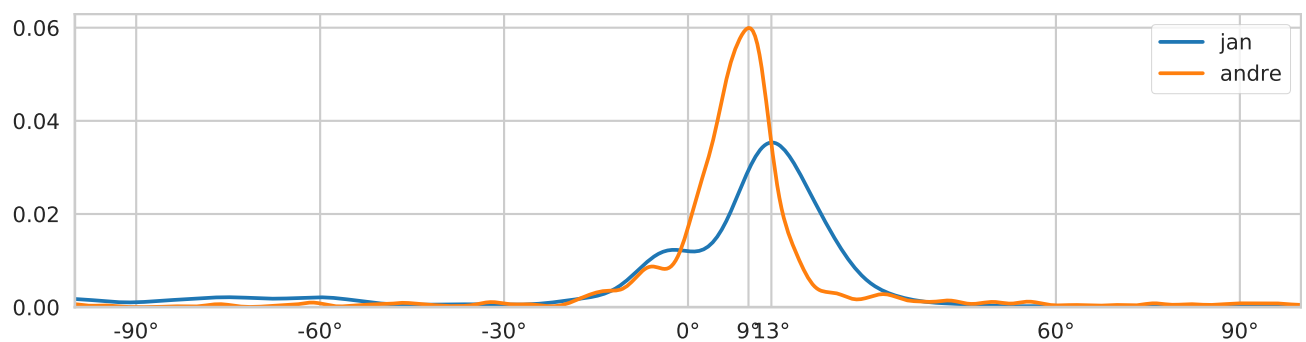
In al 4 toetse is daar 'n duidelike konstante rigtingberamingsfout. Dit is duidelik uit die feit dat die verdeling van die rigtingberamingsfoute in Figure 10.23(b), 10.24(b), 10.25(b) en 10.26(b) geeneen 'n gemiddeld van 0° het nie. Omdat die oriëntasie van die opnemers met die blote oog geskat is, is die konstante rigtingfout te wagte. Vir toekomstige toetse is dit wenslik om 'n laser of akkurate kompas te gebruik om die apparate te oriënteer.

Veelsymmetrie is ook uitgevoer vir elkeen van die bogenoemde toetse. Die resultaat van toets D word in Figuur 10.27 getoon. Die ander resultate is soortgelyk. Soos verwag het dit nie gewerk nie, omdat die 6 mikrofone van elke opnemer baie naby aan mekaar is in vergelyking met die afstand tussen die opnemers en die klankbron, sodat die 6 mikrofone as't ware as 'n enkele mikrofoon beskou kan word (in §10.5.4 bespreek). Dit veroorsaak 'n sleggeaarde matriks in Vergelyking 10.22, wat veroorsaak dat die oplossings vir die klankbronposisie almal naby 'n lyn loodreg op die lyn tussen die opnemers lê. Dit stem ooreen met die simulasiresultate in Figuur 10.19 en die wiskundige verduideliking in Bylae B.3.

¹⁹Breë-band klanke veroorsaak duideliker pieke in kruiskorrelasies, en bied dus oor die algemeen meer akkurate rigtingberamings.

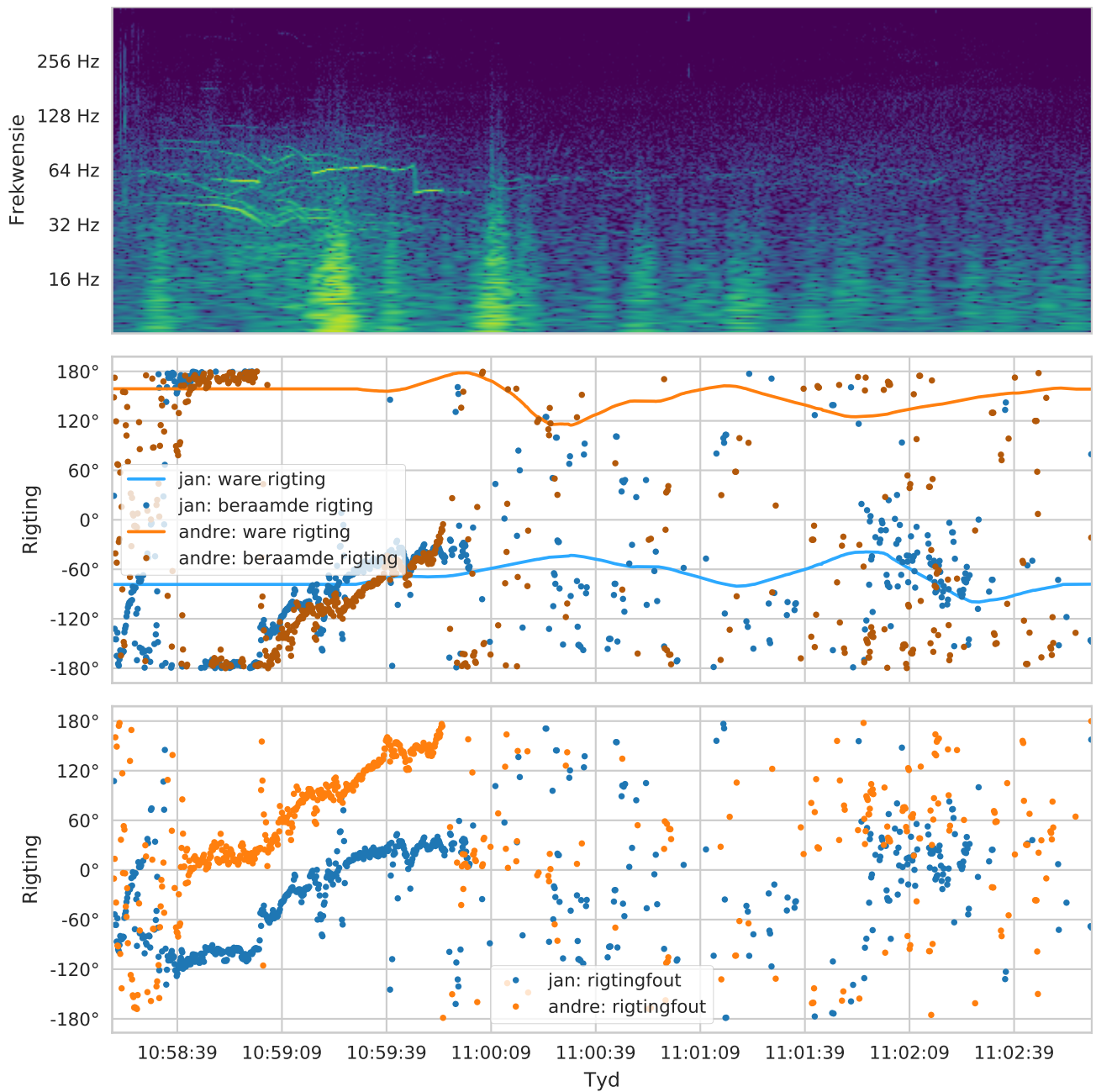


(a) Van bo na onder: Spektrogram van een van die mikrofoonseine tydens hierdie toets; Rigtingberamings; Rigtingberamingsfoute. Die tyd-as word gedeel.

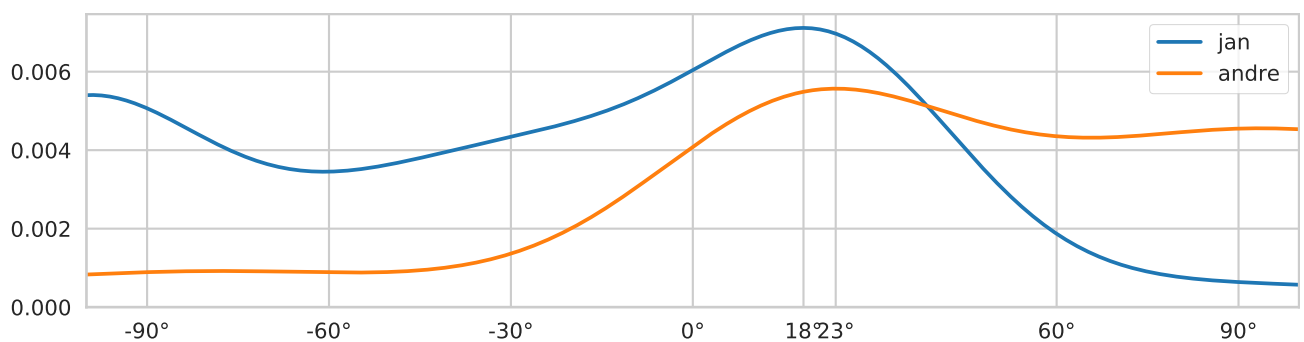


(b) Kerndigtheidsberaming van rigtingfoute.

Figuur 10.24: Rigtingberamingsfoute per opnemer vir toets B, met naby-gespasieerde appaarte en 'n olifantdreungeluide as bronsin.

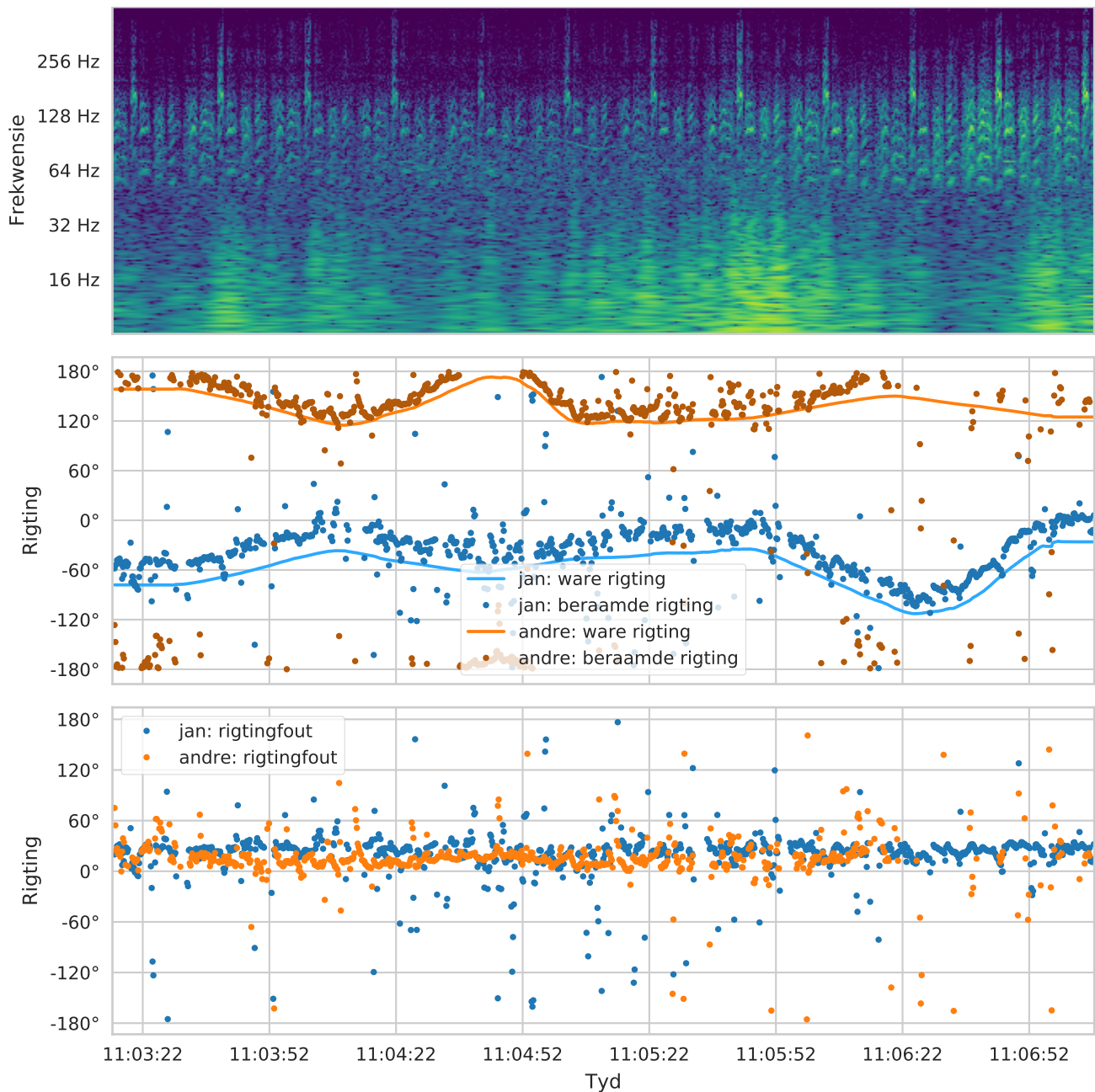


(a) Van bo na onder: Spektrogram van een van die mikrofoonseine tydens hierdie toets; Rigtingberamings; Rigtingberaamingsfoute. Die tyd-as word gedeel.

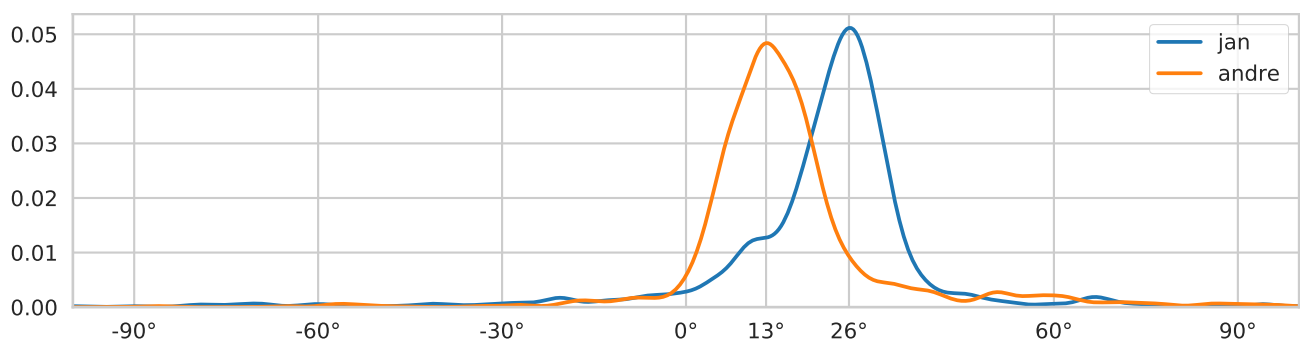


(b) Kerndigtheidsberaming van rigtingfoute.

Figuur 10.25: Rigtingberamingsfoute per opnemer vir toets C, met wyd-gespasieerde apparate en 'n ruisbronsin.

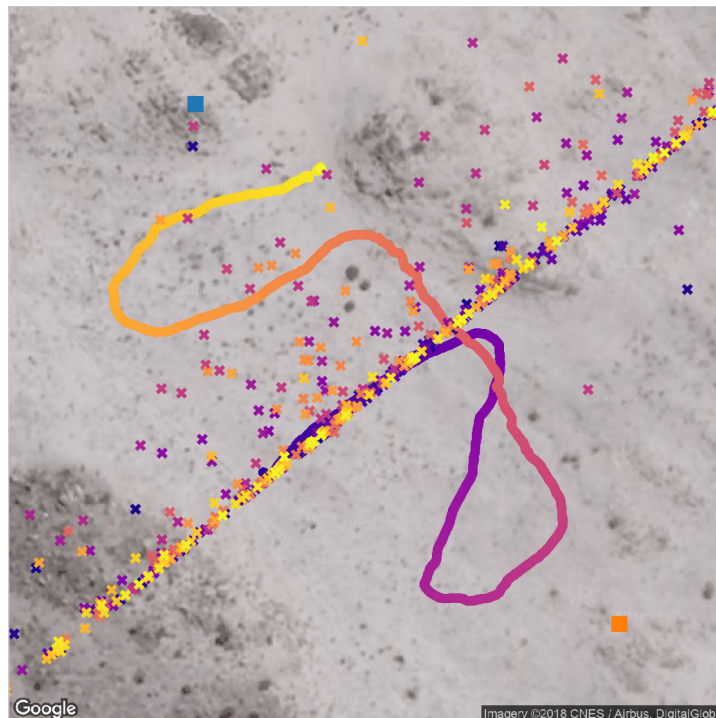


(a) Van bo na onder: Spektrogram van een van die mikrofoonseine tydens hierdie toets; Rigtingberamings; Rigtingberaamingsfoute. Die tyd-as word gedeel.



(b) **Kerndigtheidsberaming** van rigtingfoute. Die posisie van die piek van elke **kerndigtheidsberaming** is waarskynlik gelyk aan die oriëntasiefout van elke opnemer. Die beraming van klankbronrigtings kan dus verbeter word deur eenvoudig hierdie waardes af te trek.

Figuur 10.26: Rigtingberamingsfoute per opnemer vir toets D, met naby-gespasieerde apparate en olifantdreungeluide as bronsein.



Figuur 10.27: Die resultaat van veelsymeting vir toets D. Die blou en oranje blokkies stel die klankopnemers voor. Die lyn stel die werklike roete voor, terwyl die kleurverandering van pers na geel die verloop van tyd voorstel. Die kruisies van dieselfde kleure is die posisieberamings met behulp van veelsymeting by elke tydstep. Soos verwag uit die simulaties in §10.5.4 en die bewys in Bylae B.3, is twee klankopnemers nie genoeg om veelsymeting toe te pas nie. Dit bly 'n belowende algoritme wat in die toekoms beter getoets moet word as daar nog van die apparaat wat in hierdie dokument ontwerp is, vervaardig word.

Hoofstuk 11

Gevolgtrekking

11.1 Opsommend

Die doel van hierdie projek was om die ligging van wilde olifante in hul natuurlike habitat te beraam, sonder enige interaksie met die dier self. Die oorsig oor die navorsingsgeskiedenis van olifantgeluide in §1.1 het getoon dat olifantdreungeluide sterk infrasoniese komponente het, en dat hul habitat, veral snags en in droë tye, ideale toestande bied vir die propagering van hul dreungeluide oor lang afstande. Alhoewel daar al ondersoek ingestel is na die herkenning van hierdie olifantdreungeluide, het daar tot dusver nog nie 'n geïntegreerde hardewarestelsel vir die opname en analise van hierdie geluide ontstaan nie. Dié tekortkoming is in hierdie tesis aangespreek. Die huidige skrywer hoop dat die ontwerpte prototipe 'n nuttige bydrae sal lewer tot sommige van die wildbewingstoepassings wat in §1.2 bespreek is. Die infrasoniese klankopnemer is uitsonderlik omdat dit die enigste outomatiseerbare oplossing tot op hede is wat nie vereis dat enige toerusting aan individuele olifante vasgemaak word nie. Die projek voldoen aan die ontwerpvereistes wat in §1.3 en §2 bespreek is: Dit *a)* is goedkoop en tog sensitief by lae frekwensies, *b)* is battery-aangedrewe, *c)* het 'n lae intrinsieke ruis (alhoewel daar ruimte vir verbetering is), *d)* kan draadloos kommunikeer via **WiFi** en *e)* kan akkuraat tydhou met behulp van **GPS**.

In §2 is bestaande klanktoerusting geassesseer. Geen klankopnemer is gevind wat aan die genoemde vereistes voldoen nie. Dit het gedien as motivering vir die daaropvolgende hoofstukke (§3–§9) waarin 'n eie klankopnemer ontwerp is. Die ontwerp van hierdie klankopnemer het die hoofsaak van hierdie tesis gevorm, en bied 'n fondasie vir verdere ondersoek in die veld van geoutomatiseerde olifantopsoring.

Geskikte kragbronne is in §3 ontwerp wat poog om hoë-stroom 5 V (§3.2) en lae-ruis 5 V, 3 V en 1,8 V (§3.3) vanaf 'n 12 V verseëlde loodsuurbattery aan die res van die stroombane te voorsien. 'n Kragbronbeheerder is in §3.5 ontwerp wat verseker dat die battery nie gedreineer word wanneer die apparaat afgeskakel word nie. In §3.6 is die werksverrigting en ruisprofiel van hierdie kragbronne met 'n **RPI** as 'n aktiewe las ondersoek, en voorstelle vir verbetering is gemaak.

In §4 is **Invensense ICS-40300**-mikrofone vir die opneem van olifantdreungeluide gekies. Dit is die enigste geskikte mikrofoon wat beide *a)* sensitief is by lae frekwensies en *b)* binne die finansiële bereik van hierdie projek is. In §4.4 is daar bevind dat windbeskerming noodsaaklik is, en dat goedkoop materiale, soos namaakpels, 'n geskikte oplossing bied. In §4.5 is daar op 'n sirkelvormige mikrofoonskikking besluit, sodat die klankbronopsoring in alle rigtings dieselfde sal wees.

Daarna, in §5, is die ontwerp van 'n oudiokoppelvlak wat by die **ICS-40300**-mikrofone pas, bespreek. Dit sluit versyferaars (§5.1), voorversterkers (§5.2) en **teen-oorvleuelingsfilters** (§5.3) in. Die oudiokoppelvlak het 'n hoë **intree**-impedansie, 'n hoogafsnifyfrekwensie van ongeveer 1 kHz, 'n monsterebreedte van 24 bisse per kanaal, 'n monsterefrekwensie van 4800 Hz, en 6 onafhanklike

kanale wat sinkroon gemonster word. Lae-toleransie komponente word gebruik om die variansie van die gedrag van die onderskeie kanale laag te hou.

Omdat die versyferaar slegs een monster op 'n slag in geheue hou, en die **RPI** nie 'n intydse stelsel is nie, was dit nodig om 'n digitale buffer tussenin te plaas. Hierdie buffer is in §6 ontwerp. Daar is bevind dat 'n vinnige mikrobeheerder met twee **DMA**-gedrewe **SPI** kanale benodig word sodat die verwerker nooit langer as 'n enkele monsterperiode besig is en sodoende te laat met die versyferaar kommunikeer nie (§6.1–6.3). 'n Eie protokol en drywer vir kommunikasie tussen die mikrobeheerder en die **RPI** via **SPI** is ontwerp (§6.6). Sagteware is geskryf om dit te implementeer op 'n wyse wat verseker dat monsters nie verlore sal gaan nie. Daarvoor is die konsep van 'n draadbestande sirkelbuffer bespreek en geïmplementeer (§6.7). Hierdie bufferimplementasie vermy die gebruik van wedersydse uitsluitings wat verwerkingstyd mors of kwasi-atomiese bewerkings wat die hantering van onderbrekingsversoeke vertraag, terwyl dit steeds gelyktydige toegang deur aparte haal- en stootdrade ondersteun. Die prioriteite van die **DMA**-kanale en onderbrekings word gestel om die kanse op verlore monsters te minimeer (§6.8–6.9). Die mikrobeheerder se versyferaar word ook gebruik om 'n spanningsmonitor te implementeer wat die battery teen totale ontlading beskerm (§6.10) en seker maak dat die **RPI** veilig afgeskakel word.

Die **RPI** wat die data prosesseer en berg, is in §7 bespreek. Noukeurige aandag is aan die hardware- en sagtewareopstelling van die **Linux** stelsel geskenk (§7.2–7.4, 7.8–7.17). Sagteware is ontwikkel wat met die buffer kommunikeer, fouttoestande hanteer, en monsters na die **SD**-kaart skryf (§7.5–7.7).

In §8 is die opstelling van 'n **GPS**-module vir die sinkronisasie van monsterring oor lang afstande bespreek. Die **Linux**-programme genaamd **GPSD** en **NTPD** word gebruik om **GPS**-data via **UART** te ontvang en die **RPI** se klok daarvolgens te stel. Die **RPI** stel op sy beurt weer die sekondeteller op die buffermikrobeheerder wanneer dit verkeerd is. Die mikrobeheerder gebruik die **PPS**-sein van die **GPS**-module om die sekondeteller aan te skuif, terwyl breukdele van 'n sekonde met behulp van die versyferaar-klok aangeteken word. Alhoewel die tydstempels wat deur die **GPS** gegee word akkuraat is tot op enkele nanosekondes, is daar 'n bewys gegee dat die maksimum sinkronisasiefout tussen die opnemers steeds een monsterperiode plus enkele nanosekondes is (§8.6). Hier is dus nog ruimte vir verbetering.

Die vervaardiging van 'n onderstel om al die verskillende komponente te huisves, is in §9 beskryf.

Om die prototipe te assesser, is verskeie algoritmes vir die opsporing van olifantdreungeluide met hierdie stelsel in §10.1–10.5 bespreek. Simulasies is uitgevoer, en toetse is gedoen met 'n luidspreker wat ruis en olifantdreungeluide speel (§10.6).

Alhoewel die ontwerpde apparaat nie direk op olifante getoets is nie, is die toetsresultate met 'n luidspreker belowend. Die opsporing van olifantdreungeluide met infrasoniese klankopnemers is dus 'n haalbare oplossing vir verskeie wildebewaringsprobleme. Die volgende afdeling beveel werk aan wat nog gedoen moet word om van hierdie prototipe 'n produk te maak wat vir wildebewaarders of bewarings-ekoloë bruikbaar sal wees.

11.2 Toekomstige werk

11.2.1 Elektroniese ruis

As gevolg van aktiewe laste soos die mikrobeheerder en **RPI** is daar ruis wat op al die toevoerspannings in die apparaat sigbaar is (Sien Figuur 3.12). Dit is waarskynlik die oorsaak van die relatief hoë versyferingsruis wat in Figuur 3.15 getoon is. Om die volle monsterbreedte van die 24-bis versyferaar te kan benut, moet die intrinsieke ruis van die klankopnemer aansienlik verlaag word. 'n Laer ruisvlak sal ook tot gevolg hê dat sagter klanke met groter detail opgeneem kan word, sodat olifante oor langer afstande opgespoor kan word. Die volgende oplossings kan ondersoek word om die ruisvlak te verlaag:

- Verbeter die kragbronontwerp. Beter lyn- en lasregulasie sal die ruis verminder. Onderzoek verskillende kragbrontopologieë, of koop 'n klaargemaakte, 'stil' kragbron wat spesifiek vir ruis-sensitiewe toerusting gemaak is.
- Gebruik versyferaars en operasionele versterkers met beter kragbron-ruisverwerping.
- Beperk die ruis tot die digitale dele van die stroombaan deur **ferrietkerne** op strategiese plekke te plaas.
- Isoleer die laste beter deur groter of beter **afvlakkapasitors** by elke kragbron en elke las te plaas.
- Gebruik koperplate wat aan **GND** gekoppel is (d.w.s. Faraday-hokke) om die analoog-stroombane teen ruis wat deur die res van die stroombaan uitgestraal word, te beskerm.
- Indien geen van die bogenoemde voorstelle daarin slaag om die versyferingsruis te verminder nie, skep dan twee aparte stroombane met twee aparte grondvlakke. Plaas die analoogkomponente en versyferaar in een stroombaan, en al die digitale komponente in die ander stroombaan. Die enigste verbinding tussen die stroombane is dan via **SPI**, en 'n digitale isoleerder (byvoorbeeld ISO7760 van **TI**) kan gebruik word om die data oor te dra sonder om die ruis oor te dra.

11.2.2 Windbeskerming

Verskillende tipes windbeskerming kan vergelyk word. Dit is moontlik dat 'n duurder tipe windbeskerming tog 'n verbetering in die **SNR** toon as die windspoed baie hoog is. Teen die betreklik lae windspoed wat aanwesig was terwyl die toetse gedoen is, is die windgeraas glad nie hoorbaar nie. Die ruisvlak in die stelsel word dus deur die elektroniese ruis bepaal, wat eerste aangespreek moet word.

11.2.3 Sinkronisasie van monsters

Tans werk die versyferaars met afsonderlike klok-kristalle, en die **GPS**-data word ná monsterring gebruik om aan elke monster 'n tydstempel toe te ken. Soos aangetoon in §8.6 kan dit soms 'n sinkronisasiefout van ongeveer een monsterperiode veroorsaak. Om hierdie probleem op te los, kan een van twee dinge probeer word:

- Verhoog die monsterfrekwensie van die versyferaar. Dit bring egter mee dat baie meer data gestoor moet word, en 'n beter versyferaar is waarskynlik nodig.
- Ontwerp 'n **GPS**-gedissiplineerde versyferaarklok. So 'n klok neem die 1 Hz **PPS**-klok en vermenigvuldig dit om 'n 4800 Hz klok vir die versyferaar te verskaf. Die klok sal waarskynlik van 'n **FPGA** of mikrobeheerder gebruik moet maak eerder as 'n gewone **PLL**, sodat logika ingebou kan word wat sal verseker dat die klok teen 'n betreklik betroubare tempo aanhou loop wanneer die **GPS**-sein tydelik sou wegval.

11.2.4 Mikrofoonafstande

Die effek van die afstande tussen die mikrofone moet ondersoek word. Die afstand tussen mikrofone in hierdie projek is 20 cm, maar dit kan vergroot word tot 34,4 cm om 'n beter hoekberamingsresolusie te kry (sien Vergelyking 10.1 en die omliggende bespreking).

11.2.5 Faseverskille

Die faseverskille tussen kanale, wat deur die hoogdeurlaatfilters in §4.2 en teen-oorvleuelingsfilters in §5.3 veroorsaak word, moet kleiner gemaak word. Die hoogdeurlaatfilters wat in §4.2 ontwerp is, is nie geïmplementeer nie. As gevolg van die nie-lineêre aard van die ontwerpsproses het dit vergete gebly, en 'n vorige weergawe met 'n afsnyfrekwensie van 16 Hz en kapasitors met 20 % toleransie is gebou. Die mikrofone met hul hoogdeurlaatfilters moet volgens Figuur 4.4 herbou word. Die faseverskille tussen die filterkanale van die teen-oorvleuelingsfilters (Sien Figuur 5.13) is heelwat kleiner as dié van die hoogdeurlaatfilters, maar kan ook verbeter word.

11.2.6 FPGA

Die sagteware op die mikrobeheerder in §6 is redelik kompleks. Die mikrobeheerder gebruik ook heelwat stroom (sien Figuur 3.2) en veroorsaak ruis in die stelsel (sien Figuur 3.12). Ondersoek kan ingestel word of so 'n buffer ook met 'n FPGA geïmplementeer kan word. Die spanningsmonitor kan dan deur 'n spanningsvergelyker ('n operasionele versterker met positiewe terugvoer en histerese) geïmplementeer word, sodat die mikrobeheerder heeltemal verwyder kan word.

11.2.7 Sagteware op die RPI

Verskeie voorstelle vir verbeterings aan die opstelling van die RPI is reeds in §7.18 gemaak, en word hier herhaal:

- Skryf gebruikersvriendelike sagteware wat die toestand van die apparaat intyds rapporteer. Dit moet
 - toon hoe vol al die betrokke buffers is,
 - waarsku as die inkomende klanksein te hard is en afgesny word,
 - die gebruiker toelaat om opnames te stop en te begin,
 - outomaties soek na USB-apparate om opnames op te stoor, en die gebruiker laat kies waar die opnames gestoor word, en
 - die gebruiker toelaat om opnames via 'n webkoppelvlak af te laai.
- Die bronkode, wat op DVD E.1 in Bylae E beskikbaar is, kan verbeter word om minder interafhanklik en meer modulêr te wees.
- Maak die oudiodata op 'n nie-blokkerende stroom (soos 'n UDP-poort) beskikbaar, sodat daar intyds na die klankseine geluister kan word vir ontfoutingsdoeleindes.
- As 'n FPGA in plaas van 'n mikrobeheerder gebruik word, behoort die SPI-kommunikasie met die RPI heelwat vinniger te kan wees, wat beteken dat DMA aan die RPI-kant gebruik kan word. Dit sal die verwerkergebruik van die program wat monsters haal (§7.7) verlaag, wat óf geleentheid bied vir meer dataverwerking op die RPI, óf die batteryleeftyd sal verbeter.
- Maak dit moontlik om die sagteware via WiFi in die veld op te dateer, sonder om die apparaat af te skakel.
- Verlaag die frekwensie van die RPI se verwerker om batterykrag te spaar indien die ekstra verwerkertyd nie nodig is nie.
- Die ionice program in Linux kan gebruik word om seker te maak dat sekere programme voor ander toegang tot stoormedia soos die SD-kaart verkry wanneer dit nodig is. ionice werk soortgelyk aan die nice en renice programme wat in §7.9 bespreek is.

11.2.8 Outomatiese, aanlyn klassifikasie

'n Klassifiseerder van olifantdreungeluide, soos die een wat in Byker (2015) en Zeppelzauer *et al.* (2014) bespreek is, kan op die RPI geïmplementeer word, tesame met die algoritmes wat in hierdie projek bespreek is. Dit skep die moontlikheid dat elke eenheid slegs rigtingberamingsinligting na 'n sentrale bediener kan stuur, waar die ligging van olifante dan intyds bepaal word.

11.2.9 Verbeter klankbronopsoring

'n Deegliker literatuurstudie oor al die verskillende klankbronopsoringsalgoritmes kan gemaak word. Meer gesofistikeerde algoritmes vir tydvertragingsberaming en klankbronopsoring bestaan. Daar moet ook spesifiek gekyk word na robuustheid teen steurseine, eggo's en weergalming, wat waarskynlik in beboste gebiede 'n groter rol speel as in die veld. Gelyktydige opsporing van meer as een olifant is ook gewens, maar nog nie aangespreek nie.

Die veelsymetring-algoritme moet deegliker getoets word deur nog GPS-gesinkroniseerde infrasoniese klankopnemers te bou, soos aanbeveel in §10.6. Voorstelle vir name van nuwe klankopnemers is 'petra', 'robin', 'monique', 'etienne', 'piek', 'lana' en 'christi'. Die ongelyksoortigheid van die name help die gebruiker om nie die toetsdata van verskillende opnemers met mekaar te verwar nie.

11.2.10 Selfstandige kragopwekking

Dit is wenslik om sonselle en 'n batterylaaier deel te maak van die projek, sodat dit langtermyn in die veld gelos kan word. Die stelsel moet dan ook teen wilde diere en weer bestand gemaak word. Die eerste stap is om die apparaat te kamoeleer, sodat diere nie van die stelsel bewus is nie. Sterker materiaal as die groen plastiek waarmee die boksies gedruk is, sal gebruik moet word. Ondersoek moet ingestel word na die wyse waarop bestaande elektroniese toerusting wat in wild-habitat gebruik word, beskerm word.

11.2.11 Ekologiese impak

As die stelsel vir langtermyn dataversameling in olifant-habitat gebruik word, is dit dalk belangrik om te toets of die teenwoordigheid van so 'n apparaat 'n negatiewe effek op die gedrag van diere het. So 'n ondersoek is waarskynlik wel buite die omvang van Elektroniese Ingenieurswese.

11.2.12 Draadlose kommunikasie

'n Langafstand radioverbinding of satelliet-internet sal die analise van ingesamelde data vergemaklik. Daar moet ook aandag gegee word aan die netwerksekuriteit op die apparaat. Veilige SSH-sleutels moet bó gewone teks-wagwoorde verkies word.

Met die bogenoemde verbeterings in plek, beloof hierdie projek om die begin van 'n haalbare oplossing vir olifantopsoring te wees.

Bylaes

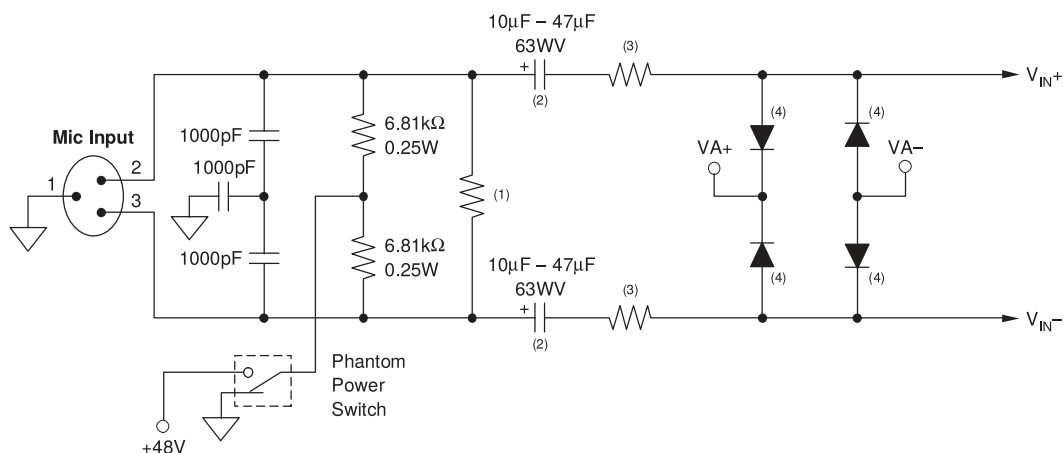
Bylae A

Die ontwerp van 'n 48V-skimkragbron vir kapasitormikrofone

Sommige mikrofone stuur 'n sein en kry kragtoevoer op dieselfde geleiers. Hierdie tipe toevoer heet **skimkrag**. Hierdie **skimkrag** is 'n **GS**-spanning (gewoonlik tussen 2 V en 48 V, afhangend van die tipe mikrofoon) wat op beide die positiewe en negatiewe lyne van 'n gebalanseerde mikrofoonkabel geplaas word om die mikrofoon se kapasitiewe membraan voor te span. Soos wat die membraan beweeg, verander die kapasitansie, en sodoende word 'n spanningsgolfvorm bo-op die **GS**-toevoerspanning gesuperponeer. Die **GS**-komponent word dan aan die ontvangkant uitgefilter voordat die **WS**-sein versterk word.

Die datablad van die PGA2500 voorversterker (Texas Instruments, 2003, p. 12) bevat 'n goeie voorbeeld van 'n stroombaan wat gebruik kan word om 'n **GS**-afset op die mikrofoonlyn te plaas. Dit word in Figuur A.1 getoon.

Die 48 V-**skimkrag**bron was nuttig tydens die beginfase van hierdie projek, toe verskillende mikrofone oorweeg is, waaronder kapasitormikrofone, wat met **skimkrag** voorgespan moet word. Toe die kragbron in §3 herontwerp is, is hierdie kragbron egter verwyder, omdat daar in §4 besluit is om **MEMS**-mikrofone vir die finale projek te gebruik, wat 'n baie laer spanning (1,8 V) op 'n aparte geleier benodig.



- NOTES: (1) Bridging resistor, used to set the impedance seen by the microphone.
 (2) The blocking capacitor value is selected based upon the desired low frequency response.
 (3) Current-limiting resistor. Select the highest value tolerable based upon input noise requirements.
 (4) Schottky diode, selected for fast turn-on and rated for a minimum of a 10A surge current.
 Recommended device is the MBRA120LT3 from ON Semiconductor.

Figuur A.1: Skimkrag-toevoer. Geneem uit Texas Instruments (2003, p. 12)

Simbool	Beskrywing	Min.	Tipies	Maks.	Eenheid
$V_{\text{SNS,OC}}$	Spanning by ISNS-pen waarby stroomsekering intree	120	150	180	mV
I_{DD}	Stroom wat by V_{DD} -pen invloei	–	1,5	2,5	mA
I_{drive}	Maksimum stroomuitset by GDRV-pen	375	400	–	mA
V_{FB}	Verwysingspanning van fout-versterker	686	700	714	mV
$V_{\text{SS,val}}$	Spanningsval van SS-pen na die fout-versterker	–	700	–	mV
V_{BP}	Spanning van die interne lineêre spanningsreguleerder as $V_{\text{in}} > V_{\text{BP}}$	7	8	9	V

Tabel A.1: Noemenswaardige spesifikasies van TPS40210 uit die datablad (Texas Instruments, 2015).

Die oorspronklike ontwerp van die 48 V kragbron word in hierdie bylae bespreek.

A.1 Spesifikasies

Vir ontwikkelingsdoeleindes is dit wenslik om die hele stelsel van **USB** te kan bedryf. 'n Minimum **intreespanning** van effens laer as 5 V word dus vereis. Die maksimum **intreespanning** behoort dieselfde of beter te wees as vir die 5 V-bron, naamlik ongeveer 15 V.

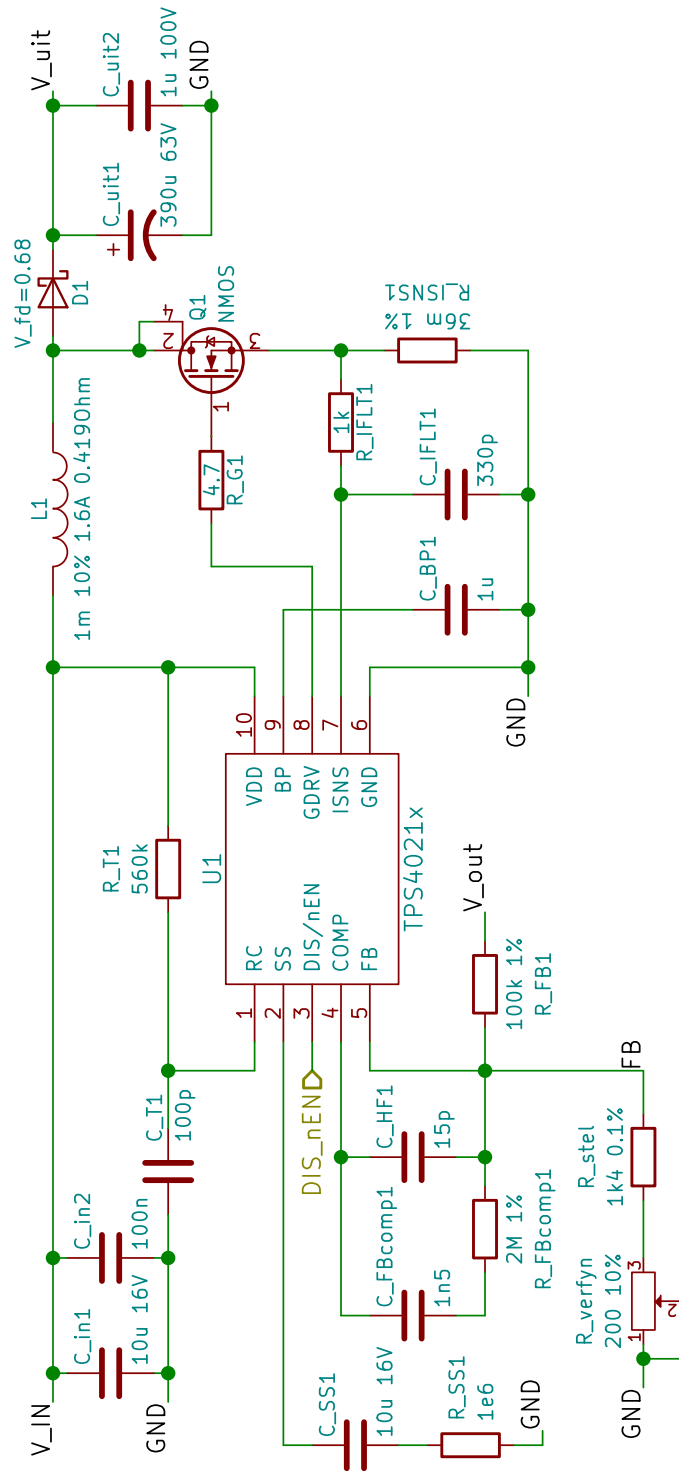
Volgens Dooley (2006) het die vroegste kapasitormikrofone wat **skimkrag** (eerder as 'n battery) benut, slegs 2 mA elk nodig gehad. Sommige nuwer mikrofone gebruik tot 10 mA (Woolf (2001), via Dooley (2006)). Die aanskakelstroom van 'n kapasitormikrofoon kan egter nog hoër wees, omdat die kapasitor nog gelaai moet word. 'n Veiligheidsfaktor van 2 word ingewerk om seker te maak dat 6 mikrofone gelyktydig aangedryf kan word. 'n **Uitsetstroom** van 120 mA teen 48 V word dus vereis. Dit beteken dat 6 mikrofone met 20 mA elk voorsien kan word. Die meeste moderne kapasitormikrofone gebruik egter heelwat minder stroom.

A.2 Ontwerp

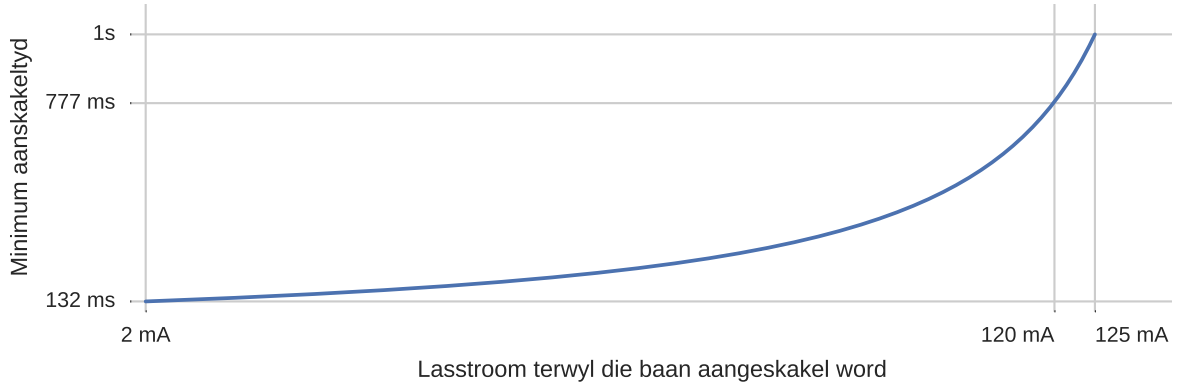
In 'n ondersoek na gepaste lae-stroom hoë-spanning kapper-reguleerders het die TPS40210 van TI uitgestaan. Tabel A.1 som die belangrikste spesifikasies van hierdie komponent op.

Die stroombaan (Figuur A.2) is aan die hand van §7–8 van die datablad ontwerp. Al die vergelykings in hierdie afdeling kom uit daardie datablad (Texas Instruments, 2015).

Daar is gepoog om die rimpelspanning onder 10 mV te hou en om die drywingseffektiwiteit minstens 85 % te maak. Die hoeveelheid ruis wat die rimpelspanning op die audio veroorsaak, is natuurlik ook afhanklik van die kragbronruisverwerping van die mikrofoon, asook die frekwensie van die rimpelspanning. Hoe verder die rimpelspanning buite die hoorbare frekwensie is, hoe beter. Die skakelfrekwensie van die kapperbaan word deur 'n RC-baan wat aan die RC-pen gekoppel is, bepaal. Vergelyking A.1 gee die verhouding tussen die RC-waardes en die skakelfrekwensie.



Figuur A.2: Stroombaandiagram van die 120mA-48V-bron uit weergawe A van die kragbron wat in §3 bespreek word. Die diagram is met KiCad ontwerp. Wanneer 'n komponent se naam nie op 'n getal eindig nie, voeg KiCad outomaties 'n 1 by. R_{T1} in hierdie figuur is byvoorbeeld dieselfde as R_T in Vergelyking A.1.



Figuur A.3: Minimum aanskakeltyd in terme van die aanvanklike lasstroom, indien die baan aangeskakel word terwyl die uittrekapasitor leeg is. As die rolle van die x- en y-asse omgeruil word, kan dit ook beskou word as die maksimum aanvanklike lasstroom vir die gegewe aanskakeltyd.

$$R_T = \frac{1}{a \cdot f_{SW} \cdot C_T + b \cdot f_{SW}^2 + c \cdot f_{SW} - d + e \cdot C_T - g \cdot C_T^2}$$

onderworpe aan $100 \Omega < R_T < 1000 \Omega$
en $68 \text{ pF} < C_T < 120 \text{ pF}$

waar R_T = weerstand in $k\Omega$,
 C_T = kapasitansie in pF ,
 f_{SW} = skakelfrekwensie in Hz ,

$$a = 5,8 \times 10^{-8}, \quad b = 8 \times 10^{-10}, \quad c = 1,4 \times 10^{-7},$$

$$d = 1,5 \times 10^{-4}, \quad e = 1,7 \times 10^{-6}, \quad \text{en } g = 4 \times 10^{-9}.$$

Met $f_{SW} = 292,5 \text{ kHz}$ en $C_T = 100 \text{ pF}$ is R_T gerieflik naby aan 'n standaard weerstandwaarde, $559,9 \text{ k}\Omega$

Die komponent skakel die uittree af as daar meer as $I_{uit,OC} = 1.2 \cdot I_{uit,maks}$ stroom vloei. Om te keer dat hierdie waarde oorskry word wanneer die baan aangeskakel word, word 'n weerstand en kapasitor tussen die SS-pen en GND gekoppel. Die tyd wat die baan neem om aan te skakel, is afhanklik van die tydkonstante van hierdie RC-netwerk.

Die verhouding tussen die aanskakeltyd en die grootte van die kapasitor word deur Vergelyking A.2a gegee. Vergelyking A.2b gee die stroom waarmee die uittrekapasitors¹ gelaai word.

$$t_{SS} = \frac{C_{SS}}{R_{SS} \cdot \ln \left(\frac{V_{BP} - V_{SS, \text{val}}}{V_{BP} - (V_{SS, \text{val}} + V_{FB})} \right)} \quad (\text{A.2a})$$

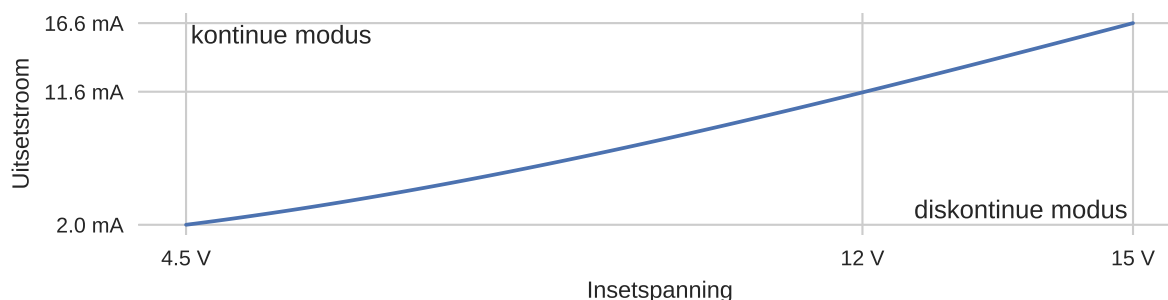
$$I_{C, \text{laai}} = \frac{C_{uit} \cdot V_{uit}}{t_{SS}} \quad (\text{A.2b})$$

waar $C_{uit} = C_{uit,1} + C_{uit,2}$

Daar is dus 'n minimum aanskakeltyd t_{SS} sodat I_{uit} minder as $I_{uit,OC}$ bly indien die baan aangeskakel word terwyl die uittrekapasitor leeg is. Figuur A.3 toon die minimum aanskakeltyd teenoor die aanvanklike uitsetstroom, vir die gegewe maksimum insetstroom. Met $R_{SS} = 1 \text{ M}\Omega$ en $C_{SS} = 10 \mu\text{F}$ is die aanskakeltyd $t_{SS} = 1 \text{ s}$ en die uittrekapasitor word met $I_{C, \text{laai}} = 18,62 \text{ mA}$ gelaai.

Die volgende komponente word deur die datablad voorgeskryf:

¹ Die waardes van die uittrekapasitors word later in die teks bepaal. Die waarde $C_{uit} = 391 \mu\text{F}$ kan reeds in Vergelyking A.2b gebruik word, omdat die ontwerp deur verskeie iterasies gegaan het.



Figuur A.4: Die grens tussen kontinue en diskontinue modusse. As die **uitsetstroom** onder 'n sekere waarde is, is daar tye waar die **uittreke** kapasitor volgelaaï word en die stroom deur die induktor dus 0 A bereik. Dit word die 'diskontinue modus' genoem.

- By die BP-pen: 'n $1\ \mu\text{F}$ kapasitor met 'n spanningshantering van ten minste 8 V.
- By die DIS-pen: 'n $1,1\ \text{M}\Omega$ aftrekweerstand.

Tydens normale werking het die komponent twee modusse, naamlik kontinue en diskontinue geleiding. Wanneer die **uitsetstroom** klein genoeg is, is daar tye wanneer die **uittreke** kapasitor volgelaaï word en die stroom deur die induktor 0 A bereik. Dit is die diskontinue modus. Vergelyking A.3² en Figuur A.4 beskryf die grens tussen die modusse in terme van die **uitsetstroom**.

$$I_{\text{uit,grens}} = \frac{(V_{\text{uit}} + V_{\text{fd}} - V_{\text{in}}) \cdot V_{\text{in}}^2}{2 \cdot (V_{\text{out}} + V_{\text{fd}})^2 \cdot f_{\text{SW}} \cdot L} \quad (\text{A.3})$$

Die **insetspanning** en **uitsetstroom** bepaal ook die **dienssiklus**. Volgens die 'tipiese karakteristieke' in die datablad moet die **dienssiklus** tussen ongeveer 7,89 % en 93,3 % wees om die skakelfrekwensie $f_{\text{SW}} = 292,5\ \text{kHz}$ te kan volhou. Figuur A.5 toon dat die **dienssiklus** tussen 8,353 % en 90,756 % sal wees vir die huidige ontwerp. In die kontinue modus is die **dienssiklus** tussen 69,187 % en 90,756 %³.

Die rimpelstroom deur die induktor bepaal die minimum induktansie en maksimum weerstand van die induktor wat nodig word. Die induktansie, die skakelfrekwensie van die baan en die **dienssiklus** bepaal weer die grootte van die rimpelstroom. Na 'n paar iterasies is 'n minimum induktansie van $303,7\ \mu\text{H}$ gevind. Toe die stroombaan gebou is, was die naaste beskikbare en bekostigbare induktor met 'n betreklik lae weerstand heelwat groter, naamlik 1 mH, met 'n ekwivalente serie-weerstand van $419\ \text{m}\Omega$. Vergelyking A.4 gee 'n benadering vir die verband tussen die rimpelstroom deur die induktor en die toevoerspanning van die stroombaan. Figuur A.6 toon hierdie verband grafies.

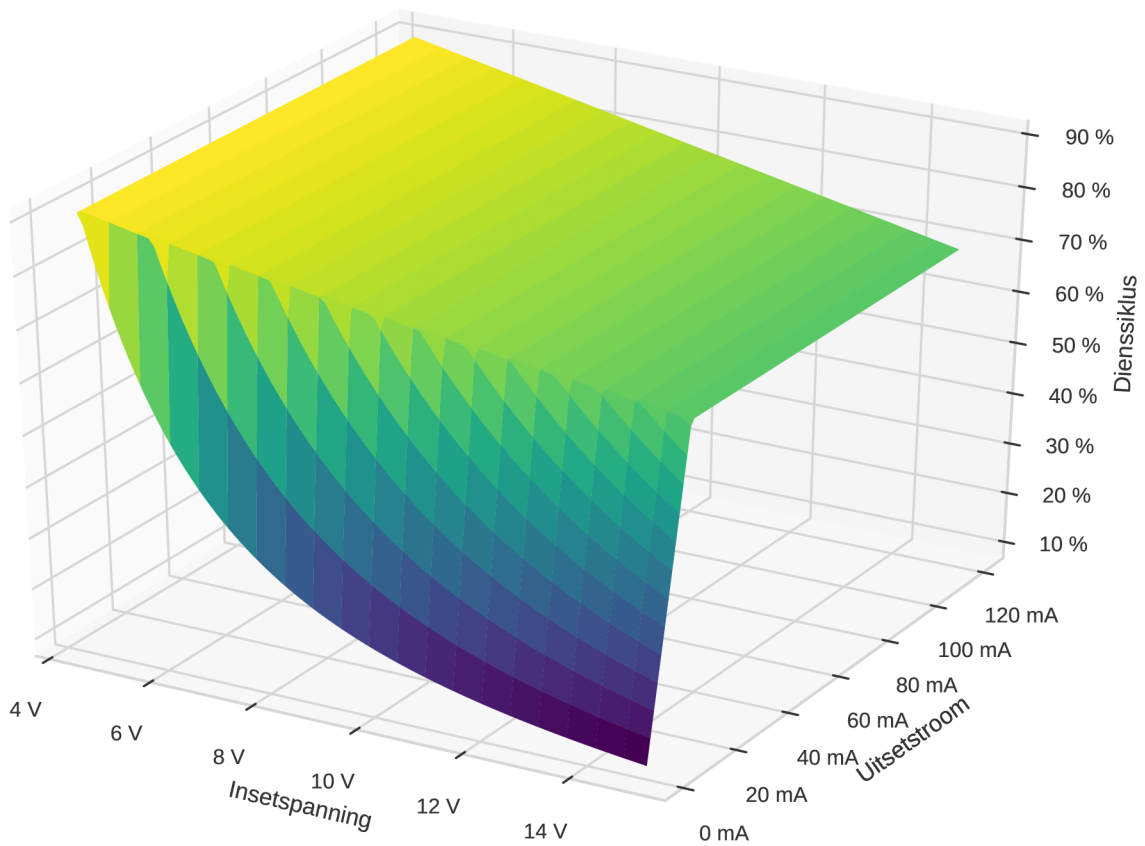
$$I_{\text{L,rimpel}} \approx \frac{V_{\text{in}} D}{L f_{\text{SW}}} \quad (\text{A.4})$$

waar D = dienssiklus.

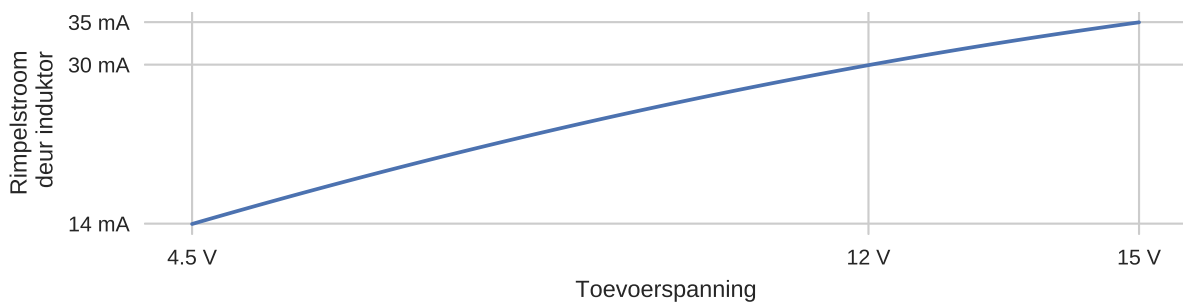
Die datablad bied twee vergelykings om die maksimum **wortel van die gemiddelde kwadraat (WKG)**-stroom (Vergelyking A.5a) en piekstroom (Vergelyking A.5b) deur die induktor te

² Die keuse van 'n Schottky-diode met voorwaartse spanning V_{fd} verskyn later in die teks. Die ontwerp het verskeie iterasies ondergaan, en daarom kan die waarde $V_{\text{fd}} = 680\ \text{mV}$ reeds in Vergelyking A.3 gebruik word.

³ Meer as die gebruiklike twee desimale plekke word benodig om afrondingsfoute in Vergelykings A.5a en A.5b te voorkom.



Figuur A.5: Die dienssiklus is 'n funksie van die insetspanning en die uitsetstroom. In die kontinue modus is die dienssiklus slegs 'n funksie van die insetspanning.



Figuur A.6: Met 'n 1 mH-induktor wissel die rimpelstroom deur die induktor van 13,962 mA by die minimum toevoerspanning van 4,5 V tot 35,480 mA by die maksimum toevoerspanning van 15 V.

benader.

$$\begin{aligned}
 I_{L,WGK} &= \sqrt{I_{L,gemiddeld}^2 + \left(\frac{I_{L,rimpel}}{12}\right)^2} \\
 &\approx \sqrt{\left(\frac{I_{uit,maks}}{1 - D_{maks}}\right)^2 + \left(\frac{I_{L,rimpel,min}}{12}\right)^2} \\
 &= \sqrt{\left(\frac{120 \text{ mA}}{1 - 90,756\%}\right)^2 + \left(\frac{13,962 \text{ mA}}{12}\right)^2} \\
 &= 1,298 \text{ A}
 \end{aligned} \tag{A.5a}$$

$$\begin{aligned}
 I_{L,peik} &\approx \frac{I_{uit,maks}}{1 - D_{maks}} + \frac{I_{L,rimpel,min}}{2} \\
 &= \frac{120 \text{ mA}}{1 - 90,756\%} + \frac{13,962 \text{ mA}}{2} \\
 &= 1,305 \text{ A}
 \end{aligned} \tag{A.5b}$$

Die gemiddelde drywingsverkwisting in die induktor word deur Vergelyking A.6 gegee.

$$P_L = I_{L,WGK}^2 L_{ESR} = (1,298 \text{ A})^2 \times 419 \text{ m}\Omega = 706,08 \text{ mW} \tag{A.6}$$

Om die **uitset**spanning gelyk te rig, word 'n diode tussen die induktor en die **uittree** benodig. 'n Schottky-diode het 'n lae voorwaartse spanningsval en verkis dus minder drywing as 'n tipiese diode. Wanneer die transistor geskakel word, verskyn daar kortliks 'n negatiewe spanning by die **metaaloksiedhalfgeleier veldeffektransistor (MOSFET, Metal-Oxide-Semiconductor Field-Effect Transistor)** se bron-terminaal. Daarom beveel die datablad aan dat die truwaartse deurbreekspanning van die diode (V_{BR}) minstens $1,25V_{uit} = 60 \text{ V}$ moet wees. Die gemiddelde stroom deur die diode is gelyk aan die maksimum **uitset**stroom en die piekstroom deur die diode is dieselfde as die piekstroom deur die induktor. 'n Schottky-diode met 'n voorwaartse spanning $V_{fd} = 680 \text{ mV}$, deurbreekspanning $V_{BR} = 100 \text{ V}$ en **stroomhantering** van $1,5 \text{ A}$ is gekies. Vergelyking A.7 gee die drywingsverkwisting in die diode by maksimum **uitset**stroom ($I_{uit} = 120 \text{ mA}$).

$$P_{D,maks} = V_{fd} I_{uit,maks} = 680 \text{ mV} \times 120 \text{ mA} = 81,6 \text{ mW} \tag{A.7}$$

Die datablad beveel aan dat elektrolitiese kapasitors in parallel met keramiekkapasitors by die **uittree** en **intree** gebruik word. Die elektrolitiese kapasitor kan meer stroom voorsien, terwyl die keramiekkapasitor 'vinniger' stroom kan voorsien (want dit het betreklik lae impedansie by hoë frekwensies). Boonop is dit moeilik om keramiekkapasitors met 'n groot genoeg kapasitansie te vind, of elektrolitiese kapasitors met 'n klein genoeg weerstand. Vergelykings A.8 toon die aanbevole minimum kapasitansies en maksimum ESR van die parallel-geskakelde kapasitors. Die gekose **uittree**kapasitors vorm saam $C_{uit} = 391 \mu\text{F}$ en $C_{uit,ESR} = 0,981 \text{ m}\Omega$. By die **intree** word 'n enkele kapasitor gebruik. Verdere kapasitor-spesifikasies word later in Tabel A.2 (p. 165) gelys.

$$C_{uit} > \frac{8 \cdot I_{uit} \cdot D}{V_{uit,rimpel} \cdot f_{SW}} \quad C_{uit,ESR} < \frac{V_{uit,rimpel}}{I_{L,peik} - I_{uit}} \tag{A.8a}$$

$$C_{in} > \frac{I_{L,rimpel}}{4 \cdot V_{in,rimpel} \cdot f_{SW}} \quad C_{in,ESR} < \frac{V_{in,rimpel}}{2 \cdot I_{L,rimpel}} \tag{A.8b}$$

Die R_{ISNS} weerstand word gebruik om te meet hoeveel stroom deur die **MOSFET** vloe. Die waarde van hierdie weerstand bepaal ook wanneer die stroomsekering intree ($I_{uit,OC}$). Volgens die datablad is daar twee beperkings op die maksimumwaarde van hierdie weerstand. Vir die beheerstelsel om stabiel te bly, skryf die datablad Vergelyking A.9a voor. Die waarde van R_{ISNS}

word verder beperk deurdat dit klein genoeg moet wees om die spanning by die ISNS-pen binne perke te hou wanneer maksimum stroom deur R_{ISNS} vloei. Vir hierdie doel word Verelyking A.9b ook in die datablad gegee.

$$R_{\text{ISNS}} < \frac{V_{\text{in}} \cdot L \cdot f_{\text{SW}}}{60 \cdot (V_{\text{uit}} + V_{\text{fd}} - V_{\text{in}})} \quad (\text{A.9a})$$

$$R_{\text{ISNS}} < \frac{V_{\text{ISNS,OC,min}}}{1,1 (I_{\text{L,piek}} + I_{\text{drive}})} \quad (\text{A.9b})$$

In die ergste geval moet $R_{\text{ISNS}} < 64 \text{ m}\Omega$. 'n $36 \text{ m}\Omega$ weerstand word gekies. Vervolgens word die maksimum drywingsverkwisting in die weerstand deur Vergelyking A.10 gegee.

$$\begin{aligned} P_{R_{\text{ISNS}}} &= I_{\text{L,WGK}}^2 R_{\text{ISNS}} D_{\text{maks}} \\ &= (1,298 \text{ A})^2 \times 36 \text{ m}\Omega \times 90,756 \% \\ &= 55,1 \text{ mW} \end{aligned} \quad (\text{A.10})$$

Onder die aanname dat daar 'n weglaatbare klein stroom by die ISNS-pen invloei, word die stroombeperking deur Vergelyking A.11 gegee. Daar is dus vir alle praktiese doeleindes geen outomatiese stroombeperking in hierdie baan nie. Die baan sal meer as 120 mA kan lewer, maar sal nie noodwendig binne die ontwerpsspesifikasies bly nie. Die teoretiese maksimum **uitset**stroom voordat iets breek, word nie hier ondersoek nie.

$$I_{\text{OC}} = \frac{V_{\text{ISNS,OC}}}{R_{\text{ISNS}}} = \frac{150 \text{ mV}}{36 \text{ m}\Omega} = 4,167 \text{ A} \quad (\text{A.11})$$

Vergelyking A.12 gee die totale drywing in die stroombaan in terme van die gewenste effektiwiteit.

$$\begin{aligned} P_{\text{totaal}} &= V_{\text{uit}} I_{\text{uit,maks}} \left(\frac{1}{\eta} - 1 \right) \\ &= 48 \text{ V} \times 120 \text{ mA} \times \left(\frac{1}{0,85} - 1 \right) \\ &= 1,016 \text{ W} \end{aligned} \quad (\text{A.12})$$

Naas die **MOSFET** is die grootste verkwisters van drywing in hierdie stroombaan die induktor, diode, R_{ISNS} en die TPS40210-skyfie self. Vergelyking A.13 gee die maksimum drywing wat die **MOSFET** mag gebruik, om steeds binne die gewenste effektiwiteit-spesifikasie te bly.

$$\begin{aligned} P_{\text{FET}} &< P_{\text{totaal}} - P_{\text{L}} - P_{\text{D,maks}} - P_{R_{\text{ISNS}}} - (V_{\text{in,maks}} I_{\text{DD,maks}}) \\ &= 1,016 \text{ W} - 706,08 \text{ mW} - 81,6 \text{ mW} - 55,1 \text{ mW} - (15 \text{ V} \times 2,5 \text{ mA}) \\ &= 136 \text{ mW} \end{aligned} \quad (\text{A.13})$$

Die datablad stel die aanname voor dat die helfte van die drywing wat in die **MOSFET** verkwis word a.g.v. skakeling is (wanneer die hek gelaai en ontlai word) – en die ander helfte a.g.v. geleiding (wanneer 'n stroom tussen die drein en die bron vloei). Vergelykings A.14 word verskaf om maksima vir die hek-bron-lading en geleidingsweerstand daar te stel.

$$Q_{\text{GS}} < \frac{3P_{\text{FET}} I_{\text{drive}}}{2V_{\text{uit}} I_{\text{uit}} f_{\text{SW}}} = \frac{3 \times 136 \text{ mW} \times 400 \text{ mA}}{2 \times 48 \text{ V} \times 120 \text{ mA} \times 292,5 \text{ kHz}} = 48,5 \text{ nC} \quad (\text{A.14a})$$

$$R_{\text{DS,aan}} < \frac{P_{\text{FET}}}{2(I_{\text{L,WGK}})^2 D_{\text{maks}}} = \frac{136 \text{ mW}}{2 \times (1,298 \text{ A})^2 \times 90,756 \%} = 44,5 \text{ m}\Omega \quad (\text{A.14b})$$

Beskikbare N-kanaal **MOSFETs** is ondersoek, en 'n 43 A, 60 V **MOSFET** met 'n geskikte hek-bron-lading $Q_{\text{GS}} = 6,7 \text{ nC}$ en maksimum geleidingsweerstand $R_{\text{DS,aan}} < 18 \text{ m}\Omega$ is gekies. Die totale heklading van hierdie **MOSFET** is $Q_{\text{G}} = 23 \text{ nC}$.

Die datablad beveel aan dat toepassings van die TPS40210 wat 'n toevoerspanning hoër as 5 V gebruik, 'n weerstand tussen die GDRV-pen en die hek van die MOSFET plaas, met 'n waarde soos in Vergelyking A.15 bereken.

$$R_G = \frac{105}{Q_G \times 10^9} = \frac{105}{23} \approx 4,7 \Omega \quad (\text{A.15})$$

Vir die passiewe filter wat deur R_{IFLT} en C_{IFLT} gevorm word, beveel die datablad die beperkings in Vergelyking A.16 aan, wat sorg dat die tydkonstante van die RC-filter 'n ordegrootte kleiner is as die pulswydte van die kapperbaan. 'n 1 k Ω -weerstand saam met 'n 330 pF-kapasitor voldoen aan hierdie vereistes.

$$1 \text{ k}\Omega < R_{\text{IFLT}} < 5 \text{ k}\Omega \quad (\text{A.16a})$$

$$\frac{C_{\text{IFLT}}}{R_{\text{IFLT}}} > \frac{0,1 D_{\text{kontinu,min}}}{f_{\text{SW}}} = \frac{0,1 \times 69,187 \%}{292,5 \text{ kHz}} \quad (\text{A.16b})$$

$$\therefore C_{\text{IFLT}} > 236,5 \times 10^{-9} R_{\text{IFLT}}$$

Die terugvoernetwerk van die uittree na die FB-pen moet aan Vergelyking A.17 voldoen. R_{FB} moet klein genoeg wees om onnodige ruis te voorkom, en groot genoeg om drywing te beperk. Die verhouding tussen R_{FB} en R_{stel} bepaal die uitsetspanning V_{uit} .

$$10 \text{ k}\Omega < R_{\text{FB}} < 100 \text{ k}\Omega \quad (\text{A.17a})$$

$$R_{\text{stel}} = \frac{V_{\text{FB}} R_{\text{FB}}}{V_{\text{uit}} - V_{\text{FB}}} \quad (\text{A.17b})$$

Om die uitreespanning by presies $V_{\text{uit}} = 48 \text{ V}$ te kry, moet daar gekompenseer word vir die toleransies van V_{FB} , R_{FB} en R_{stel} . Dit word gedoen deur 'n verstelbare weerstand in serie met R_{stel} te plaas. Met die gekose weerstande soos in Tabel A.2 (p. 165) aangeteken, sal dit altyd moontlik wees om die uittreeweerstand na presies 48 V te stel.

Die foutversterker in die TPS40210-skyfie vorm 'n beheerstelsel-lus wat met behulp van kompensasië verbeter kan word. Dit word gedoen deur 'n netwerk van kapasitors en weerstande tussen die FB- en COMP-penne (sien Figuur A.2) te verbind. Om 'n optimale beheerlus daar te stel, moet die baan eers vervaardig en geanaliseer word, omdat die resultate afhanklik is van die borduitleg. Die datablad bied egter 'n benaderde metode om die kompensasiënetwerk te ontwerp, wat voldoende is vir die huidige toepassing.

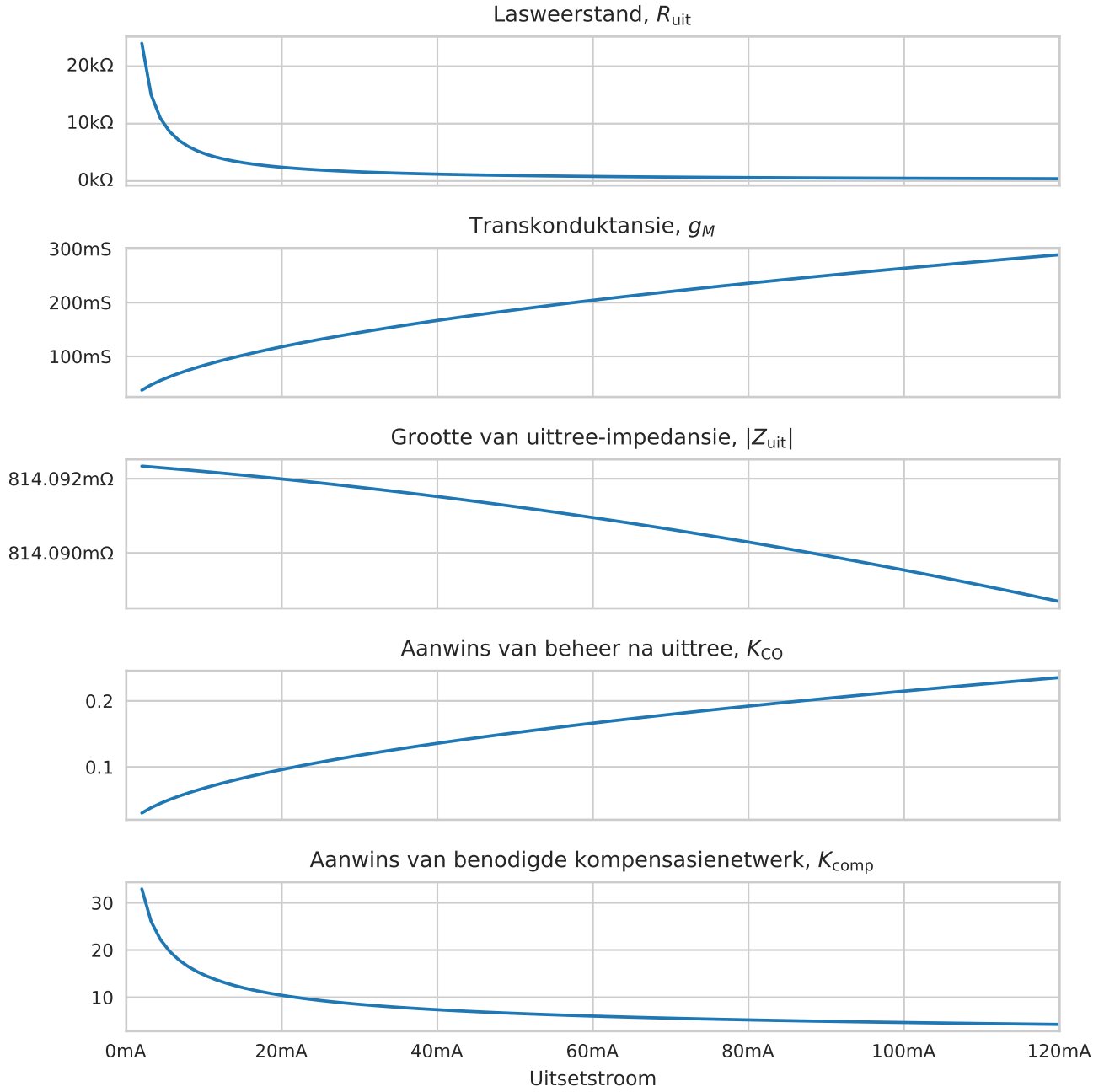
Om 'n geskikte oorgangsfrekwensie f_{CO} te kies, word inligting omtrent frekwensiegedrag van die stroom wat deur hierdie kragbron se las getrek sal word, benodig. Hierdie inligting is egter nie beskikbaar nie. 'n Voorlopige waarde van 500 Hz word gekies, met die voorneme om die waarde te hersien as die gedrag van die stelsel nie na wense is nie. 'n Lae waarde vir f_{CO} sal waarskynlik 'n stabiele stelsel met 'n stadige impulsweergawe tot gevolg hê.

Vergelykings A.18 beskryf die beheerstelselparameters van die stroombaan in terme van die uitsetstroom. Figuur A.7 beeld hierdie parameters uit.

$$R_{\text{uit}} = \frac{V_{\text{uit}}}{I_{\text{uit}}} = \frac{48 \text{ V}}{I_{\text{uit}}} \quad (\text{A.18a})$$

$$g_M = \frac{0,13 \sqrt{\frac{L f_{\text{SW}}}{R_{\text{uit}}}}}{R_{\text{ISNS}}^2 (120 R_{\text{ISNS}} + L f_{\text{SW}})} \quad (\text{A.18b})$$

$$= 0,834 \sqrt{I_{\text{uit}}}$$



Figuur A.7: Beheerstelselparameters van die TPS40210-gebaseerde opkapperkragbron.

$$|Z_{uit}|(f_{CO}) = R_{uit} \sqrt{\frac{1 + (2\pi f_{CO} C_{uit,ESR} C_{uit})^2}{1 + (R_{uit}^2 + 2R_{uit} C_{uit,ESR} + C_{uit,ESR}^2) (2\pi f_{CO} C_{uit})^2}} \quad (A.18c)$$

$$\therefore |Z_{uit}(500 \text{ Hz})| = \frac{1}{\sqrt{434 \times 10^{-6} I_{uit}^2 + 61,7 \times 10^{-6} I_{uit} + 1,5088728}}$$

$$\begin{aligned} K_{CO} &= g_M |Z_{uit}(500 \text{ Hz})| \\ &= \frac{1}{\sqrt{623 \times 10^{-6} I_{uit} + 88,6 \times 10^{-6} + 2,17 I_{uit}^{-1}}} \end{aligned} \quad (A.18d)$$

$$K_{comp} = \frac{1}{K_{CO}} = \sqrt{623 \times 10^{-6} I_{uit} + 88,6 \times 10^{-6} + 2,17 I_{uit}^{-1}} \quad (A.18e)$$

Komponent	Spesifikasies	Vervaardiger en modelnommer
D	$V_{fd} = 680 \text{ mV}$ Schottky	Vishay VS-10MQ100-M3/5AT
L	$1 \text{ mH} \pm 10 \%$, ESR = $419 \text{ m}\Omega$	Murata 1410516C
$C_{uit,1}$	$390 \mu\text{F} \pm 20 \%$, ESR = $52 \text{ m}\Omega$	Nichicon UPW1J391MHD
$C_{uit,2}$	$1 \mu\text{F} \pm 10 \%$, ESR = $1 \text{ m}\Omega$	TDK C3216X7R2A105K160AA
C_{in}	$1 \mu\text{F} \pm 10 \%$, ESR = $1 \text{ m}\Omega$	Yageo 222278113676
R_{ISNS}	$36 \text{ m}\Omega \pm 1 \%$	Panasonic ERJ6BWR036V
Q	43 A, 60 V N-MOSFET, $Q_{GS} = 6,7 \text{ nC}$, $R_{DS,aan} < 18 \text{ m}\Omega$	ON Semiconductor NTD5865NT4G
R_{IFLT}	$1 \text{ k}\Omega$	—
C_{IFLT}	330 pF	—
R_{FB}	$100 \text{ k}\Omega \pm 1 \%$	—
R_{stel}	$1,4 \text{ k}\Omega \pm 0,1 \%$	—
R_{verfyn}	$200 \Omega \pm 10 \%$	—
$R_{FB,comp}$	$2 \text{ M}\Omega$	—
$C_{FB,comp}$	$1,5 \text{ nF}$	—
C_{HF}	15 pF	—

Tabel A.2: Komponentwaardes vir die 48 V-kapperkragbron

Die aanwins K_{comp} van die kompensasie-netwerk (Vergelyking A.18e) moet die inverse van die oop-lus-aanwins K_{CO} van die beheerstelsel (Vergelyking A.18d) wees, om die eenheidsaanwins van die geslote beheerlus by die gewenste f_{CO} te plaas. Die aanwins van die beheerstelsel hang egter van die **uittree**-impedansie van die baan af, wat weer van die **uitset**stroom afhang. By $I_{uit} = 0 \text{ A}$ is 'n oneindige groot aanwins in die kompensasienetwerk nodig om die gewenste **oorgangsfrekwensie** $f_{CO} = 500 \text{ Hz}$ te behou. Die aanname word gemaak dat die mikrofone ten minste 2 mA sal gebruik. Die gewenste aanwins by minimum **uitset**stroom is dan $K_{comp} (I_{uit} = 2 \text{ mA}) = 32,9$. Vergelyking A.19 gee die gewenste waarde vir die kompensasieweerstand in terme van die terugvoerweerstand en die gewenste aanwins.

$$R_{FB,comp} = R_{FB} \times K_{comp} = 100 \text{ k}\Omega \times 32,9 = 3,29 \text{ M}\Omega \quad (\text{A.19})$$

Kies $R_{FB,comp} = 2 \text{ M}\Omega$, en plaas 'n zero een dekade onder f_{CO} (Vergelyking A.20).

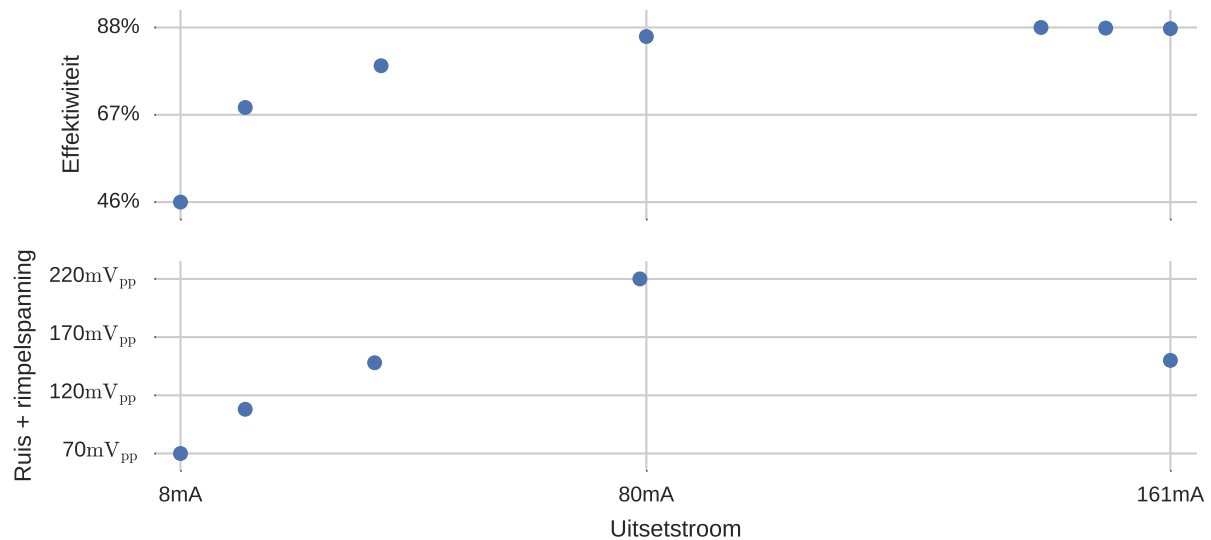
$$C_{FB,comp} = \frac{10}{2\pi f_{CO} R_{FB,comp}} = \frac{10}{2\pi \times 500 \text{ Hz} \times 2 \text{ M}\Omega} = 1,59 \text{ nF} \quad (\text{A.20})$$

Kies $C_{FB,comp} = 1,5 \text{ nF}$. Plaas 'n hoë-frekwensie pool by $5f_{CO}$, maar minder as die helfte van die eenheidsaanwins-bandwydte (minimum $1,5 \text{ MHz}$) van die foutversterker in die TPS40210 (Vergelyking A.21).

$$C_{HF} = \frac{1}{2\pi 10 f_{CO} R_{FB,comp}} = \frac{1}{2\pi \times 10 \times 500 \text{ Hz} \times 2 \text{ M}\Omega} = 15,9 \text{ pF} \quad (\text{A.21})$$

Kies $C_{HF} = 15 \text{ pF}$.

Die finale keuse van komponente word in Tabel A.2 gegee.



Figuur A.8: Gedrag van die 48 V-bron met veranderende lasstroom.

A.3 Toets

Die gedrag van die 48 V-kragbron is getoets deur verskeie diskrete weerstande in parallel te skakel en aan die bron se **uittree** te koppel. Figuur A.8 toon dat die effektiwiteit van die kragbron skerp saam met die **uitsetstroom** toeneem. Die ruis en rimpelspanning by die **uittree** is 'n minimum by lae **uitsetstroom** en bereik 'n maksimum van 220 mV_{pp} by 80 mA.

Die werklike gedrag van hierdie kragbron behoort in praktyk effens anders te wees, afhangend van die kapasitansie van die mikrofoon wat daarmee voorgespan word. Sonder gedetailleerde data oor hoe die mikrofoon se impedansie verander na aanleiding van klankgolwe, is dit onmoontlik om te bepaal hoe die kragbron die klanksein sal affekteer. Hierdie inligting omtrent mikrofone word nie gepubliseer nie en is moeilik om te meet sonder gespesialiseerde toerusting. Sulke metings is nie uitgevoer nie, omdat kapasitormikrofone (en dus hierdie kragbron) nie deel vorm van die finale ontwerp nie.

Bylae B

Wiskundige bewyse vir spesiale gevalle by veelsymetring

B.1 'n Klankbron wat op 'n simmetrie-as lê

Stelling B.1.1. *As 'n klankbron op presies een van die simmetrie-asse van 'n reghoekige mikrofoonskikking lê, is die resultaat van veelsymetring 'n lyn. Daardie lyn is gelyk aan die simmetrie-as waarop die klankbron lê.*

Bewys. Laat d_x en d_y die breedte en lengte van 'n reghoek wees. $d_x > 0$ en $d_y > 0$. Plaas vier mikrone: By $m_0 = (0; 0)$, $m_1 = (0; d_y)$, $m_2 = (d_x; 0)$ en $m_3 = (d_x; d_y)$. Plaas die klankbron by $(p_x; p_y)$ waar $p_x = \frac{d_x}{2}$ en $p_y \notin \left\{0, \frac{d_y}{2}, d_y\right\}$. Stel die mikrofoonposisies en klankbronposisies in Vergelyking 10.15 in:

$$\begin{aligned} r_0 &= \sqrt{\left(\frac{d_x}{2}\right)^2 + (p_y)^2}, & r_1 &= \sqrt{\left(\frac{d_x}{2}\right)^2 + (p_y - d_y)^2}, \\ r_2 &= \sqrt{\left(-\frac{d_x}{2}\right)^2 + (p_y)^2}, & \text{en } r_3 &= \sqrt{\left(-\frac{d_x}{2}\right)^2 + (p_y - d_y)^2}. \end{aligned}$$

Nou is $r_0 = r_2$ en $r_1 = r_3$ en $r_0 \neq r_1$. Omdat $\tau_n = \frac{r_n - r_0}{v}$, is $\tau_2 = 0$ en $\tau_1 = \tau_3$. Beskou voortaan die klankbronposisies as onbekend, en τ_n as bekend vir alle n .

Met $\tau_2 = \tau_0$ (en dus $r_2 = r_0$) in Vergelyking 10.20a is $m_{2,x}^2 + m_{2,y}^2 = 2m_{2,x}p_x + 2m_{2,y}p_y$, sodat $p_x = \frac{d_x}{2}$.

Daar is dus geen oplossing vir p_y nie, en $p_x = \frac{d_x}{2}$ is die simmetrie-as waarop die klankbron lê. \square

Hierdie bewys is rotasie-invariant. As die reghoek nie parallel met die x- en y-asse lê nie, kan die asse geroteer word. Dieselfde bewys sal geld.

B.2 'n Klankbron wat op twee simmetrie-asse lê

Stelling B.2.1. *As 'n klankbron op die middelpunt (die kruising van die simmetrie-asse) van 'n reghoekige mikrofoonskikking lê, is die resultaat van veelsymetring 'n punt. Daardie punt is die middelpunt van die reghoek.*

Bewys. Laat d_x en d_y die breedte en lengte van 'n reghoek wees. $d_x > 0$ en $d_y > 0$. Plaas vier mikrone: By $m_0 = (0; 0)$, $m_1 = (0; d_y)$, $m_2 = (d_x; 0)$ en $m_3 = (d_x; d_y)$. Plaas die klankbron by $(p_x; p_y)$ waar $p_x = \frac{d_x}{2}$ en $p_y = \frac{d_y}{2}$. Stel die mikrofoonposisies en klankbronposisies in

Vergelyking 10.15 in:

$$\begin{aligned} r_0 &= \sqrt{\left(\frac{d_x}{2}\right)^2 + \left(\frac{d_y}{2}\right)^2}, & r_1 &= \sqrt{\left(\frac{d_x}{2}\right)^2 + \left(-\frac{d_y}{2}\right)^2}, \\ r_2 &= \sqrt{\left(-\frac{d_x}{2}\right)^2 + \left(\frac{d_y}{2}\right)^2}, & \text{en } r_3 &= \sqrt{\left(-\frac{d_x}{2}\right)^2 + \left(-\frac{d_y}{2}\right)^2}. \end{aligned}$$

Nou is $r_0 = r_1 = r_2 = r_3$. Omdat $\tau_n = \frac{r_n - r_0}{v}$, is $\tau_1 = \tau_2 = \tau_3$. Beskou voortaan die klankbronposisies as onbekend, en τ_n as bekend vir alle n .

Met $\tau_1 = \tau_0$ (en dus $r_1 = r_0$) in Vergelyking 10.20a is $m_{1,x}^2 + m_{1,y}^2 = 2m_{1,x}p_x + 2m_{1,y}p_y$, sodat $p_y = \frac{d_y}{2}$. Soortgelyk, met $\tau_2 = \tau_0$ (en dus $r_2 = r_0$) is $p_x = \frac{d_x}{2}$.

Die oplossing is dus presies dieselfde as die klankbronposisie, wat ook die middelpunt van die reghoek is. \square

Hierdie bewys is rotasie-invariant. As die reghoek nie parallel met die x- en y-asse lê nie, kan die asse geroteer word. Dieselfde bewys sal geld.

B.3 Die effek van kort mikrofoonafstande

Stelling B.3.1. *As die afstande tussen enige twee mikrofone weglaatbaar klein word, benader daardie twee mikrofone 'n enkele mikrofoon, in die sin dat die resultaat van veelsymeting ekwivalent is aan die resultaat wat verkry sou word as slegs een van daardie mikrofone teenwoordig sou wees.*

Bewys. Laat d_x en d_y die breedte en lengte van 'n reghoek wees. $d_x > 0$ en $d_y > 0$. Plaas vier mikrofone: By $m_0 = (0; 0)$, $m_1 = (0; d_y)$, $m_2 = (d_x; 0)$ en $m_3 = (d_x; d_y)$. Plaas die klankbron by $(p_x; p_y)$ waar $p_x \notin \{0, \frac{d_x}{2}, d_x\}$ en $p_y \notin \{0, \frac{d_y}{2}, d_y\}$ en $p_x \neq \pm p_y$, sodat die waardes van τ_n almal verksillend is.

Uit Vergelyking 10.15:

$$\begin{aligned} r_0 &= \sqrt{(p_x)^2 + (p_y)^2} \\ r_1 &= \sqrt{(p_x)^2 + (p_y - d_y)^2} \\ r_2 &= \sqrt{(p_x - d_x)^2 + (p_y)^2} \\ r_3 &= \sqrt{(p_x - d_x)^2 + (p_y - d_y)^2} \\ \therefore \lim_{d_x \rightarrow 0} r_2 &= r_0 \quad \text{en} \quad \lim_{d_x \rightarrow 0} r_3 = r_1 \end{aligned}$$

Omdat $\tau_n = \frac{r_n - r_0}{v}$ is $\lim_{d_x \rightarrow 0} \tau_2 = 0$ en $\lim_{d_x \rightarrow 0} \tau_3 = \tau_1$. \square

Bylae C

Rugsteunkopieë van SD-kaarte

Om die voortgang van die projek te vergemaklik, is **rugsteunkopieë** van die **SD**-kaart van elke apparaat gemaak. Hierdie **afbeeldings** is op DVD **E.5** in Bylae **E** beskikbaar. Om die **SD**-kaarte **af te beeld**, of die **afbeeldings** na 'n **SD**-kaart te herstel, word 'n **Linux**-rekenaar met 'n **SD**-kaartleser en die volgende pakkette benodig:

- **pigz**
- **partclone**
- **util-linux**
- **parted** (weergawe 3.2 of nuwer)
- **e2fsprogs**

Alles kan soos volg met **apt** installeer word:

```
sudo apt install pigz partclone util-linux parted e2fsprogs
```

Die meeste van die opdragte in die volgende twee afdelings benodig **sudo**. Omdat hierdie opdragte direk met die stoormediums werk, moet die gebruiker daarop let dat die korrekte apparaatnaam gebruik word. 'n Geringe tikfout (byvoorbeeld **/dev/sda** in plaas van **/dev/sda2**) kan veroorsaak dat die **partisietabel** van 'n stoormedium saam met alle data wat daarop is, vernietig word.

Vir die voorbeelde hieronder word daar aangeneem dat die **SD**-kaart by **/dev/mmcblk0** te vind is, en die naam van die opnemer waarvoor 'n rugsteunkopie gemaak word, 'andre' is.

C.1 Beeld die SD-kaart af

Om 'n **afbeelding** van die **SD**-kaart te maak, kan die volgende stappe uitgevoer word:

- Haal **SD**-kaart uit die **RPI** en sit dit in die rekenaar.
- Maak seker dat die kaart nêrens in gebruik is nie:

```
umount /dev/mmcblk0p1
umount /dev/mmcblk0p2
```

- Verwyder alle lêers in **/home/pi/recordings** en **/var/log/locations** op die tweede **partisie** van die **SD**-kaart:

```
mkdir -p tmp_mount
mount /dev/mmcblk0p2 tmp_mount
rm -f tmp_mount/home/pi/recordings/*
rm -f tmp_mount/var/log/location/*
# As die umount-bevel te vinnig kom, is die
# SD-kaart nog besig om rm uit te voer. Wag.
sleep 1
umount tmp_mount
rmdir tmp_mount
```

- d) Toets vir foute in die ext4-lêergids:

```
e2fsck -f /dev/mmcblk0p2
```

- e) Krimp die lêergids tot 'n minimum. Alhoewel dit met partclone nie die grootte van die afbeelding-lêer affekteer nie, bepaal dit wel die minimum grootte van die SD-kaart waarheen dit herstel kan word. Die afbeeldings kan na SD-kaarte van 4 GB of groter herstel word, al is 32 GB-kaarte oorspronklik gebruik.

```
resize2fs -M -p /dev/mmcblk0p2
```

Die lêergids is nou kleiner as die partisie. Dis nie nou nodig om die partisie ook kleiner te maak nie. Nadat partclone uitgevoer is, kan die lêergids weer vergroot word om die partisie te vul.

- f) Gebruik partclone om 'n rugsteunkopie van die lêergids in elke partisie te maak. Dit is baie meer effektief as om dd te gebruik, omdat dit leë spasies oorslaan en dus 'n baie kleiner afbeelding-lêer maak. Pers dit met behulp van pigz saam.

```
partclone.fat16 --clone --source /dev/mmcblk0p1 --logfile \
YYYYMMDD_andre_p1.partclone.log | pigz -c --best \
> YYYYMMDD_andre_p1.partclone.img.gz
partclone.ext4 --clone --source /dev/mmcblk0p2 --logfile \
YYYYMMDD_andre_p2.partclone.log | pigz -c --best \
> YYYYMMDD_andre_p2.partclone.img.gz
```

- g) Maak 'n rugsteunkopie van die partisietabel:

```
sfdisk --dump /dev/mmcblk0 > YYYYMMDD_andre_partitiontable.sfdisk.dump
```

- h) Opsioneel, indien die kaart weer gebruik gaan word:

- a) Maak seker dat partisie 2 so groot as moontlik is:

```
parted /dev/mmcblk0 resizepart 2 -- -1s
```

- b) Maak die lêergids net so groot soos die partisie:

```
resize2fs /dev/mmcblk0p2
```

- c) Toets weer vir foute:

```
e2fsck -f /dev/mmcblk0p2
```

d) Maak seker dat alle veranderinge na die SD-kaart geskryf is:

```
sync
```

e) Plaas die SD-kaart in die RPI terug.

'n Skrip wat al die bogenoemde stappe uitvoer is op DVD E.5 in Bylae E beskikbaar. Hier volg 'n voorbeeld van die `stdout` wanneer dit uitgevoer word:

```
Are you sure that device /dev/mmcblk0 is the SD card from the RPI that you want
to backup? This backup script will modify the partition table! [y/n]y

Unmounting /dev/mmcblk0p1 and /dev/mmcblk0p2

umount: /dev/mmcblk0p1: not mounted
umount: /dev/mmcblk0p2: not mounted

Mounting /dev/mmcblk0p2 at tmp_mount and removing recordings and location files

Checking filesystem on /dev/mmcblk0p2

e2fsck 1.42.13 (17-May-2015)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/mmcblk0p2: 87160/1915424 files (0.7% non-contiguous), 726736/7774592 blocks

Shrinking filesystem on /dev/mmcblk0p2

resize2fs 1.42.13 (17-May-2015)
Resizing the filesystem on /dev/mmcblk0p2 to 797762 (4k) blocks.
Begin pass 2 (max = 14193)
Relocating blocks          XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Begin pass 3 (max = 238)
Scanning inode table       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The filesystem on /dev/mmcblk0p2 is now 797762 (4k) blocks long.

Backing up /dev/mmcblk0p1 to 20180502_jan_p1.partclone.img.gz

Partclone v0.2.86 http://partclone.org
Starting to clone device (/dev/mmcblk0p1) to image (-)
Reading Super Block
Elapsed: 00:00:01, Remaining: 00:00:00, Completed: 100.00%
Total Time: 00:00:01, 100.00% completed!
done!
File system:  FAT16
```

```
Device size: 66.1 MB = 129024 Blocks
Space in use: 24.0 MB = 46880 Blocks
Free Space: 42.1 MB = 82144 Blocks
Block size: 512 Byte
Elapsed: 00:00:02, Remaining: 00:00:00, Completed: 100.00%, Rate: 720.08MB/min,
current block: 129024, total block: 129024, Complete: 100.00%
Total Time: 00:00:02, Ave. Rate: 720.1MB/min, 100.00% completed!
Syncing... OK!
Partclone successfully cloned the device (/dev/mmcblk0p1) to the image (-)
Cloned successfully.
```

```
Backing up /dev/mmcblk0p2 to 20180502_jan_p2.partclone.img.gz
```

```
Partclone v0.2.86 http://partclone.org
Starting to clone device (/dev/mmcblk0p2) to image (-)
Reading Super Block
Elapsed: 00:00:01, Remaining: 00:00:00, Completed: 100.00%
Total Time: 00:00:01, 100.00% completed!
done!
File system: EXTFS
Device size: 3.3 GB = 797762 Blocks
Space in use: 2.5 GB = 618771 Blocks
Free Space: 733.1 MB = 178991 Blocks
Block size: 4096 Byte
Elapsed: 00:02:14, Remaining: 00:00:00, Completed: 100.00%, Rate: 1.13GB/min,
current block: 794621, total block: 797762, Complete: 100.00%
Total Time: 00:02:14, Ave. Rate: 1.1GB/min, 100.00% completed!
Syncing... OK!
Partclone successfully cloned the device (/dev/mmcblk0p2) to the image (-)
Cloned successfully.
```

```
Backing up partition table of /dev/mmcblk0 to
20180502_jan_partitiontable.sfdisk.dump
```

```
Enlarging partition /dev/mmcblk0p2
```

```
Information: You may need to update /etc/fstab.
```

```
Enlarging filesystem on /dev/mmcblk0p2
```

```
resize2fs 1.42.13 (17-May-2015)
Resizing the filesystem on /dev/mmcblk0p2 to 7774592 (4k) blocks.
The filesystem on /dev/mmcblk0p2 is now 7774592 (4k) blocks long.
```

```
Checking filesystem on /dev/mmcblk0p2
```

```
e2fsck 1.42.13 (17-May-2015)
Pass 1: Checking inodes, blocks, and sizes
```



```

Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/mmcblk0p2: 87160/1915424 files (0.7% non-contiguous), 726736/7774592 blocks

Syncing...

Done!

```

C.2 Herstel die afbeelding na die SD-kaart

Om 'n **afbeelding** na 'n **SD-kaart** te herstel, kan die volgende stappe uitgevoer word:

- a) Sit 'n **SD-kaart** van ten minste 4 GB in die rekenaar.
- b) Maak seker dat die kaart nêrens in gebruik is nie:

```

for p in /dev/mmcblk0*
do
    umount $p
done

```

- c) Skryf 'n nuwe **partisietabel** na die **SD-kaart**:

```

sfdisk /dev/mmcblk0 < YYYYMMDD_andre_partitiontable.sfdisk.dump

```

- d) Herstel die lêergidse na die **SD-kaart**. Hierdie bevel kan lank by 'Syncing..' (sinkroniseer) bly hang. Dit is 'n normale gevolg van die gebruik van kasgeheue en die feit dat **Linux** soms die fisiese skryf van data na stoormediums uitstel.

```

zcat 20180502_andre_p1.partclone.img.gz | partclone.fat16 --restore \
    --output /dev/mmcblk0p1
zcat 20180502_andre_p2.partclone.img.gz | partclone.ext4 --restore \
    --output /dev/mmcblk0p2

```

- e) Maak seker dat **partisie 2** so groot as moontlik is:

```

parted /dev/mmcblk0 resizepart 2 -- -1s

```

- f) Maak die **lêergids** net so groot soos die **partisie**:

```

resize2fs /dev/mmcblk0p2

```

- g) Toets vir foute:

```

e2fsck -f /dev/mmcblk0p2

```

- h) Maak seker dat alle veranderinge na die **SD-kaart** geskryf is:

```
| sync
```

i) Plaas die **SD**-kaart in 'n **RPI**.

'n **Skip** wat al die bogenoemde stappe uitvoer is op DVD **E.5** in Bylae **E** beskikbaar. 'n Voorbeeld van die **stdout** wanneer dit uitgevoer word, volg:

```
Are you sure that device /dev/mmcblk0 is the SD card to which you want to restore
the backup? All data on the casd will be destroyed! [y/n]y

Unmounting all partitions of /dev/mmcblk0

umount: /dev/mmcblk0: not mounted

Writing new partition table to /dev/mmcblk0

Checking that no-one is using this disk right now ... OK

Disk /dev/mmcblk0: 30 GiB, 32227983360 bytes, 62945280 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x20b3d1be

Old situation:

Device          Boot  Start      End  Sectors  Size Id Type
/dev/mmcblk0p1          8192  137215  129024    63M  c W95 FAT32 (LBA)
/dev/mmcblk0p2     137216 6895615 6758400   3.2G 83 Linux

>>> Script header accepted.
>>> Script header accepted.
>>> Script header accepted.
>>> Script header accepted.
>>> Created a new DOS disklabel with disk identifier 0x20b3d1be.
Created a new partition 1 of type 'W95 FAT32 (LBA)' and of size 63 MiB.
/dev/mmcblk0p2: Created a new partition 2 of type 'Linux' and of size 14.4 GiB.
/dev/mmcblk0p3:
New situation:

Device          Boot  Start      End  Sectors  Size Id Type
/dev/mmcblk0p1          8192  137215  129024    63M  c W95 FAT32 (LBA)
/dev/mmcblk0p2     137216 30318591 30181376 14.4G 83 Linux

The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

Restoring /dev/mmcblk0p1 from 20180502_andre_p1.partclone.img.gz
```

```
Partclone v0.2.86 http://partclone.org
Starting to restore image (-) to device (/dev/mmcblk0p1)
Calculating bitmap... Please wait... done!
File system:  FAT16
Device size:   66.1 MB = 129024 Blocks
Space in use:  24.0 MB = 46864 Blocks
Free Space:    42.1 MB = 82160 Blocks
Block size:    512 Byte
Elapsed: 00:00:02, Remaining: 00:00:00, Completed: 100.00%, Rate: 719.83MB/min,
current block: 129024, total block: 129024, Complete: 100.00%
Total Time: 00:00:02, Ave. Rate: 719.8MB/min, 100.00% completed!
Syncing... OK!
Partclone successfully restored the image (-) to the device (/dev/mmcblk0p1)
Cloned successfully.
```

```
Restoring /dev/mmcblk0p2 from 20180502_andre_p2.partclone.img.gz
```

```
Partclone v0.2.86 http://partclone.org
Starting to restore image (-) to device (/dev/mmcblk0p2)
Calculating bitmap... Please wait... done!
File system:  EXTFS
Device size:   3.3 GB = 793484 Blocks
Space in use:  2.5 GB = 622504 Blocks
Free Space:    700.3 MB = 170980 Blocks
Block size:    4096 Byte
Elapsed: 00:02:08, Remaining: 00:00:00, Completed: 100.00%, Rate: 1.20GB/min,
current block: 793472, total block: 793484, Complete: 100.00%
Total Time: 00:02:08, Ave. Rate: 1.2GB/min, 100.00% completed!
Syncing... OK!
Partclone successfully restored the image (-) to the device (/dev/mmcblk0p2)
Cloned successfully.
```

```
Enlarging partition /dev/mmcblk0p2
```

```
Information: You may need to update /etc/fstab.
```

```
Enlarging filesystem on /dev/mmcblk0p2
```

```
resize2fs 1.42.13 (17-May-2015)
Resizing the filesystem on /dev/mmcblk0p2 to 7851008 (4k) blocks.
The filesystem on /dev/mmcblk0p2 is now 7851008 (4k) blocks long.
```

```
Checking filesystem on /dev/mmcblk0p2
```

```
e2fsck 1.42.13 (17-May-2015)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
```

```
Pass 5: Checking group summary information
/dev/mmcblk0p2: 87824/1931520 files (0.8% non-contiguous), 731479/7851008 blocks

Syncing...

Done!
```

Bylae D

Installeer sagteware op 'n nuwe bedryfstelsel

Indien die gebruiker verlang om die sagteware van hierdie projek op 'n nuwe of ander weergawe van Linux te installeer, kan die volgende stappe gevolg word. Dit behoort egter voldoende te wees om een van die afbeeldings op DVD E.5 in Bylae E te gebruik, die instruksies in Bylae C.2 te volg, en daarna net die naam van die RPI te verander.

- a) Laai 'n ligte Linux-weergawe wat vir die RPI bedoel is, soos Emlid Raspbian RT¹ of Raspbian Lite², af.
- b) Maak seker dat die kaart nêrens in gebruik is nie:

```
for p in /dev/mmcblk0*
do
    umount $p
done
```

- c) Skryf die afbeelding direk na die SD-kaart:

```
pv emlid-raspbian-20160718.img > /dev/mmcblk0
sync
```

- d) Lig die bedryfstelsel in dat die partisietabel verander het:

```
partprobe /dev/mmcblk0
```

- e) Toets vir foute:

```
e2fsck -f /dev/mmcblk0p2
```

- f) Nuwer weergawes van Raspbian vergroot outomaties die tweede partisie en lêergids. Dit neem enkele minute wanneer die RPI vir die eerste keer aangeskakel word met daardie SD-kaart. Indien 'n ouer weergawe van Raspbian gebruik word, of iets soos Emlid Raspbian RT, volg die volgende stappe³:

¹ Weergawe 20160718 is in hierdie projek gebruik, en is dus deeglik getoets. Beskikbaar by <https://emlid.com/raspberry-pi-real-time-kernel/>

² Dit is vinnig getoets, en dit lyk of dit ook werk. Gewone Raspbian is heelwat groter en bevat baie onnodige programme wat waarskynlik 'n negatiewe impak op die opname-program sal hê. Beskikbaar by <https://www.raspberrypi.org/downloads/raspbian/>

³ As hierdie stappe met die gewone Raspbian uitgevoer word, blyk dit dat die RPI vries as dit die eerste keer aangeskakel word.

- i) Maak seker dat **partisie 2** so groot as moontlik is:

```
parted /dev/mmcblk0 resizepart 2 -- -1s
```

- ii) Maak die **lêergids** net so groot soos die **partisie**:

```
resize2fs /dev/mmcblk0p2
```

- iii) Toets weer vir foute:

```
e2fsck -f /dev/mmcblk0p2
```

- g) Maak seker dat alle veranderinge na die **SD**-kaart geskryf is:

```
sync
```

- h) Maak 'n tydelike plek waar die **lêergidse** gemonteer kan word:

```
mkdir -p tmp_mount
```

- i) Kopieer die sagteware van hierdie projek⁴ na die **SD**-kaart:

```
mount /dev/mmcblk0p2 tmp_mount
cp -r rpi_driver tmp_mount/home/pi/
cp -r rpi_config tmp_mount/home/pi/
sleep 1
umount tmp_mount
```

- j) In nuwer weergawes van Raspbian is die **SSH**-diens aanvanklik afgeskakel. Maak seker dat die **SSH**-diens geaktiveer word wanneer die bedryfstelsel vir die eerste keer **afgeskop** word deur 'n lêer met die naam **ssh** in die lêergids van die eerste partisie te plaas⁵:

```
mount /dev/mmcblk0p1 tmp_mount
touch tmp_mount/ssh
sleep 1
umount tmp_mount
```

- k) Maak seker dat alle veranderinge na die **SD**-kaart geskryf is:

```
sync
```

- l) Plaas die **SD**-kaart in die **RPI** en skakel dit aan. Die gewone Raspbian-stelsel vergroot outomaties die tweede **partisie** om die **SD**-kaart te vul. Die Emlid-stelsel doen dit nie, maar dit kan maklik met behulp van die **raspi-config** nutsprogram gedoen word, of soortgelyke stappe as in Bylae C.2 kan gevolg word voordat die kaart na die **RPI** oorgeplaas word.

- m) Maak seker dat die **RPI** aan 'n netwerk gekoppel is waar dit 'n **IP**-adres met behulp van 'Dynamic Host Configuration Protocol' (**DHCP**) sal kry. As so 'n netwerk nie beskikbaar is nie (byvoorbeeld op dié van die Universiteit Stellenbosch, waar elke rekenaar eers geregistreer moet word voordat 'n **IP**-adres daaraan toegeken sal word), kan 'n tweede netwerkkaart in enige rekenaar met internettoegang geplaas word. Koppel dan die **RPI** aan daardie netwerkpoort, en kies 'Shared to other computers' in die 'NetworkManager' program wat saam met die meeste **Ubuntu**-agtige **Linux**-weergawes versprei word.

⁴ Beschikbaar op DVD E.1 in Bylae E.

⁵ Sien <https://www.raspberrypi.org/documentation/remote-access/ssh/>

- n) Vind die **IP** van die **RPI** met 'n program soos 'Angry IP Scanner'.
- o) Indien hierdie **RPI** voorheen met 'n ander **SD**-kaart via **SSH** bereik is, sal die **SSH**-program die waarskuwing wys: 'REMOTE HOST IDENTIFICATION HAS CHANGED'. In daardie geval moet die vorige inligting omtrent hierdie **RPI** eers verwyder word:

```
ssh-keygen -f "/home/dolf/.ssh/known_hosts" -R 10.42.0.34
```

- p) Maak seker dat die wagwoord⁶ nie weer gevra word nie:

```
ssh-copy-id pi@10.42.0.34
```

- q) Teken op die **RPI** se bevelreël in:

```
ssh pi@10.42.0.34
```

- r) Gebruik **raspi-config** om die naam van die **RPI** te verander.

- s) Korrigeer gebruikersregte:

```
sudo chown -R pi:pi *
cd rpi_config
find . -type f -name "*.sh" -exec chmod +x "{}" \;
```

- t) (Opsioneel) Neem die **uitset** van die volgende bevel in 'n tekslêer op:

```
script ~/install.log
```

- u) Indien die **RPI** se datum ver van die huidige datum is, stel dit reg om probleme met **https** te voorkom:

```
date +%Y%m%d -s "20180508"
```

- v) Installeer die sagteware. Kyk uit vir foute, en druk Y soos nodig. As dit klaar is, druk **CTRL+D** om die opname te stop.

```
./install.sh
```

- w) Herbegin die bedryfstelsel. Dit apparaat behoort nou gereed te wees vir gebruik.

```
sudo reboot
```

⁶Die standaardwagwoord is: **raspberrypi**

Bylae E

DVDs

Hierdie bylae bevat DVDs met die volgende inligting:

DVD E.1 Bronkode, simulasies en allerlei.

- **misc**
 - **kicad** (KiCad-lêers wat gebruik of geskep is vir die hardware-ontwerp in hierdie projek)
 - **parts_ordered** (Volledige geskiedenis van alle parte wat bestel is sedert die begin van 2016)
 - **uml** (Bronkode vir verskeie UML-diagramme, waarvan sommige in die tesis te siene is)
- **repositories**
 - **2015_skripsie** (Die repositoria uit die skripsie wat hierdie projek voorafgegaan het)
 - * **2015_pe448_annotations**
 - * **2015_pe448_sourcecode**
 - * **2015_pe448_verslag**
 - **grootvoet_device**
 - * **hardware**
 - **buffer** (Die mikrobeheerderbordjie)
 - **chassis** (Die AutoDesk Inventor-projek vir die onderstel wat die apparaat huisves)
 - **gps** (Die stroombaanbordjie wat die **GPS**-module aan die **RPI** koppel)
 - **microphone** (Die mikrofoonstroombaanbord en hoogdeurlaatfilter)
 - **on_off** (Die kragbronbeheerder)
 - **power** (Die kragbronne)
 - **preamp_filter_adc** (Die oudiokoppelvlak)
 - * **software**
 - **control_panel** (Verskeie programmetjies wat via **SSH** met die **RPIs** kommunikeer. Handig vir toets-dae.)
 - **pic32_buffer** (MPLAB-X-projek waarmee die PIC32-mikrobeheerder geprogrammeer is)
 - **processing** (Allerlei **skripte** en kodes, waarvan die meeste tydelik van aard was. Processing is 'n verwarrende naam, want die meeste van die kode wat verwerking doen, is onder **simulasies** of op die DVDs met toetsdata, DVDs **E.2** en **E.3**.)

- `rpi_config` (Die instruksies wat die **RPI** opstel om te werk soos dit moet vir hierdie projek)
- `rpi_driver` (Die C-kode wat die **RPI** en die buffer met mekaar laat gesels via SPI)
- `python-packages`
 - * `cart-converters` (Sien <https://github.com/rudolfbyker/cart-converters>)
 - * `grootvoet` (Python-biblioteek vir seinverwerking, beide in simulaties en toetse)
 - * `lpsd` (Sien <https://github.com/rudolfbyker/lpsd>)
 - * `venus` (Sien <https://github.com/rudolfbyker/venus6>)
- `simulasies` (Simulasiekodes en resultate)
- `thesis_tex` (Die \LaTeX -bronkode van die tesis)

DVDs E.2 en E.3 Toetsdata en -kode. Hierdie DVDs bevat 'n argief van alle toetsdata, -kode en -resultate. Die `README.md` lêer verduidelik hoe om by die data uit te kom. Kortom:

```
cat toetse.tar.xz.* | unxz -c | tar -xf -
```

DVD E.4 Foto's en video's.

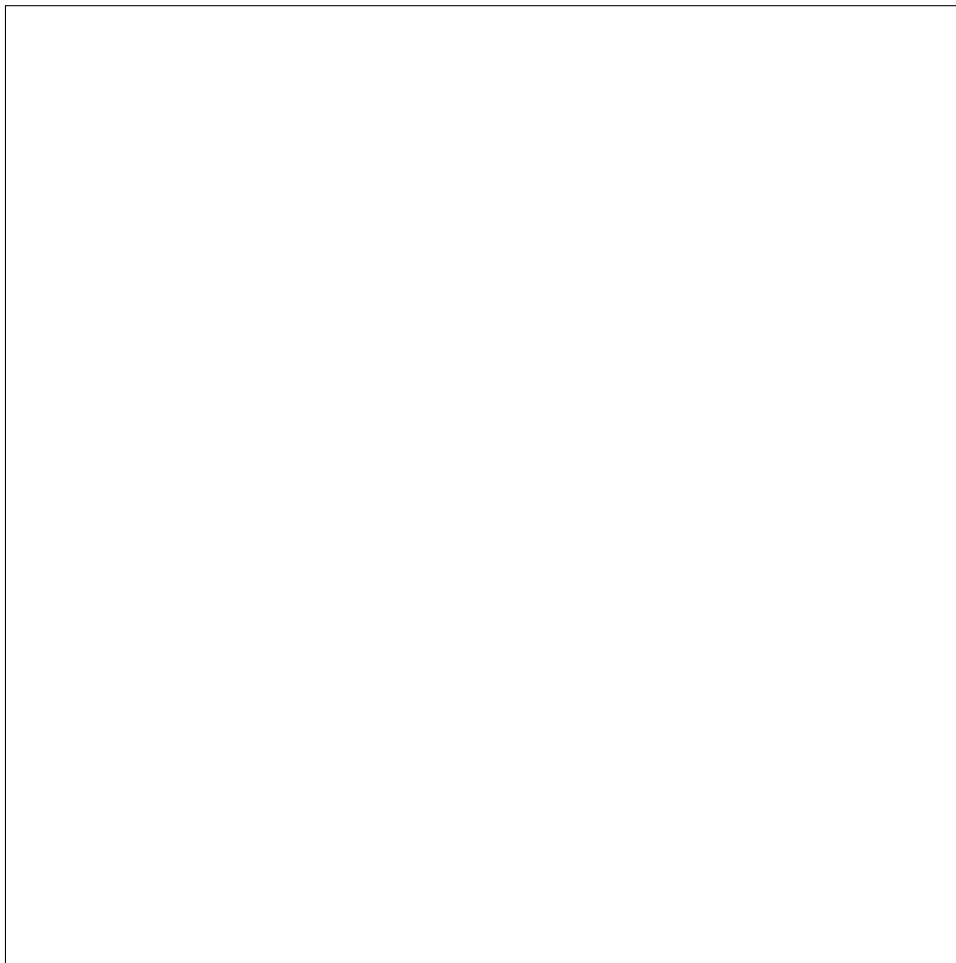
- `toetse` (Foto's en video's van spesifieke toetsuitstappies)
 - `20171222_jurgen_coetzenburg_doa` (Saam met Jürgen Lüdemann by Coetzenburg)
 - `20180206_jono_coetzenburg` (Saam met Jonathan Wotherspoon by Coetzenburg)
 - `20180220_verskeie` (Alleen, op verskeie plekke)
 - `20180224_kaappunt` (Saam met Andries Stephanus Lourens Heystek in die Kaapse Skiereiland)
- `fotos` (Ander foto's)
- `videos` (Ander video's)

DVD E.5 SD-kaart-afbeeldings. Hierdie DVD bevat die **afbeeldings** en **skripte** waarna daar in Bylae C verwys word.

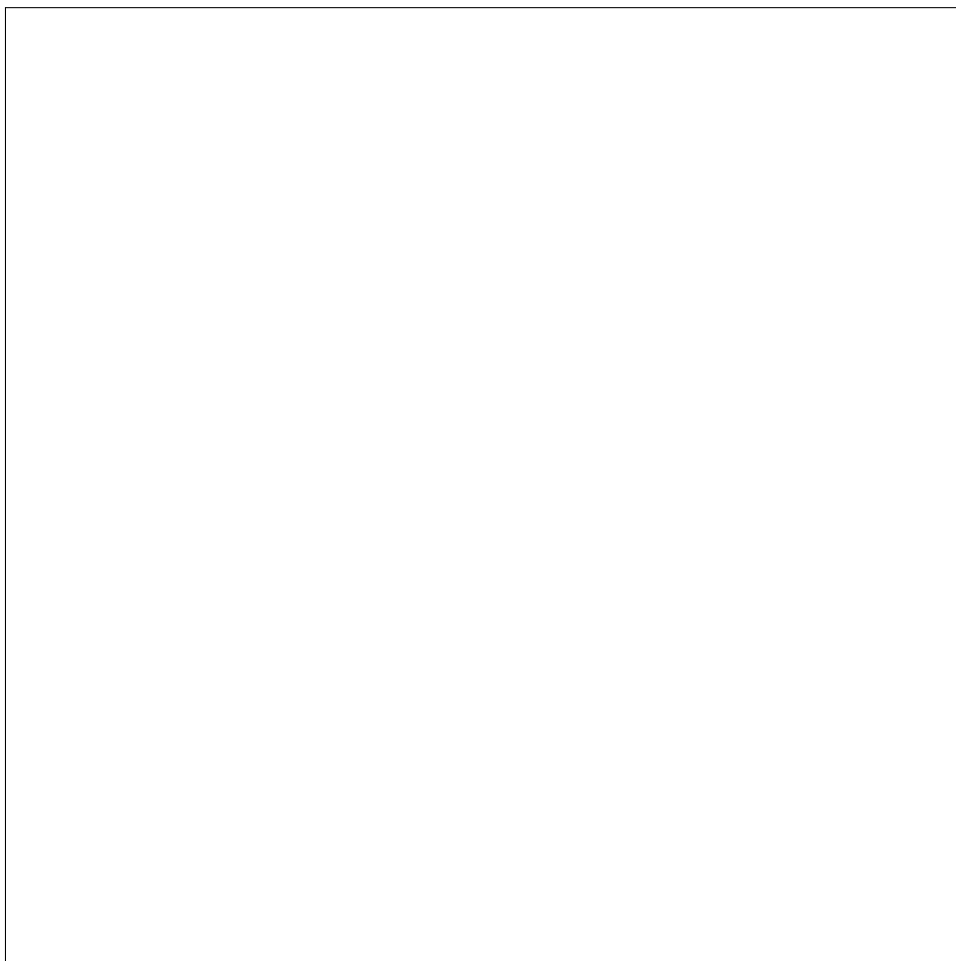
Hierdie DVDs is op 'n Linux-rekenaar vervaardig met behulp van die volgende bevel:

```
genisoimage -r -f -o dvd.iso /pad/na/gids/
wodim dvd.iso
```

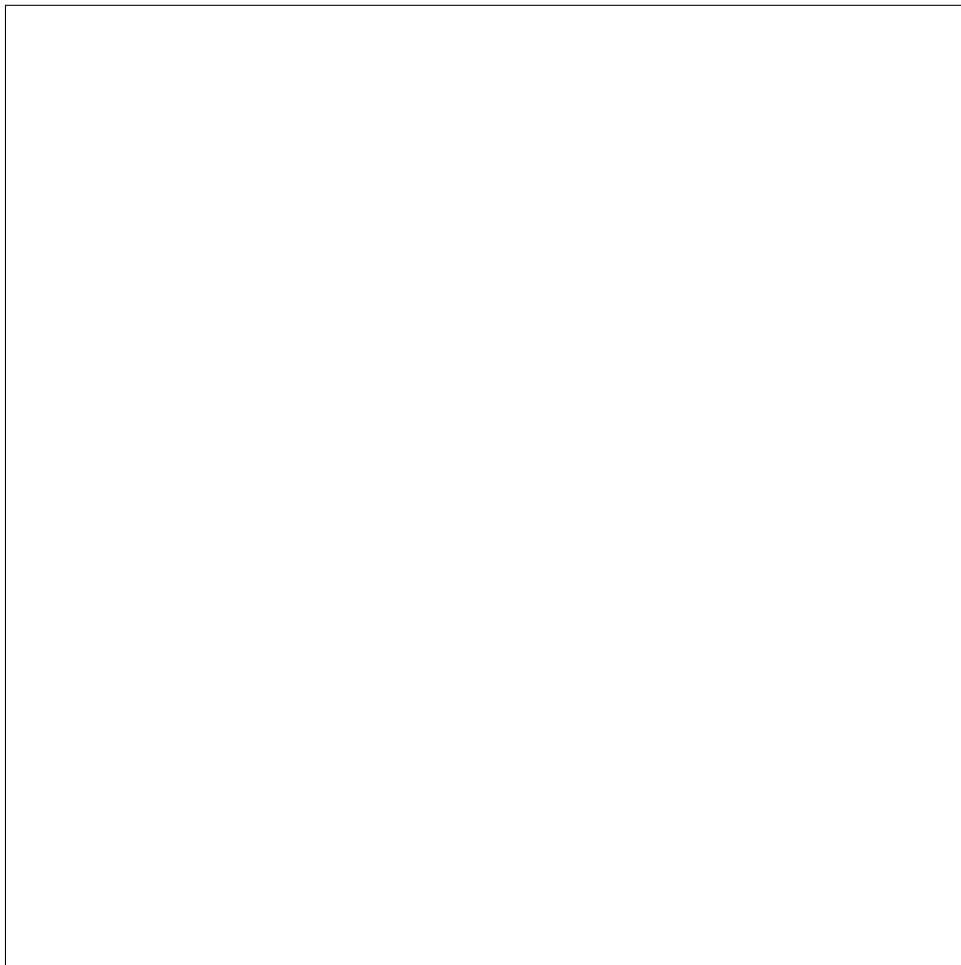
Dit skep 'n data-DVD met 'n ISO9660 lêergids met die Rock Ridge-uitbreiding, wat lang lêername vir Linux-stelsels bewaar. Microsoft Windows-bedryfstelsels kan egter nie die Rock Ridge-data lees nie, en verkort alle lêername in die 8.3-formaat. Gebruik dus slegs in 'n Linux-rekenaar om hierdie DVDs te lees.



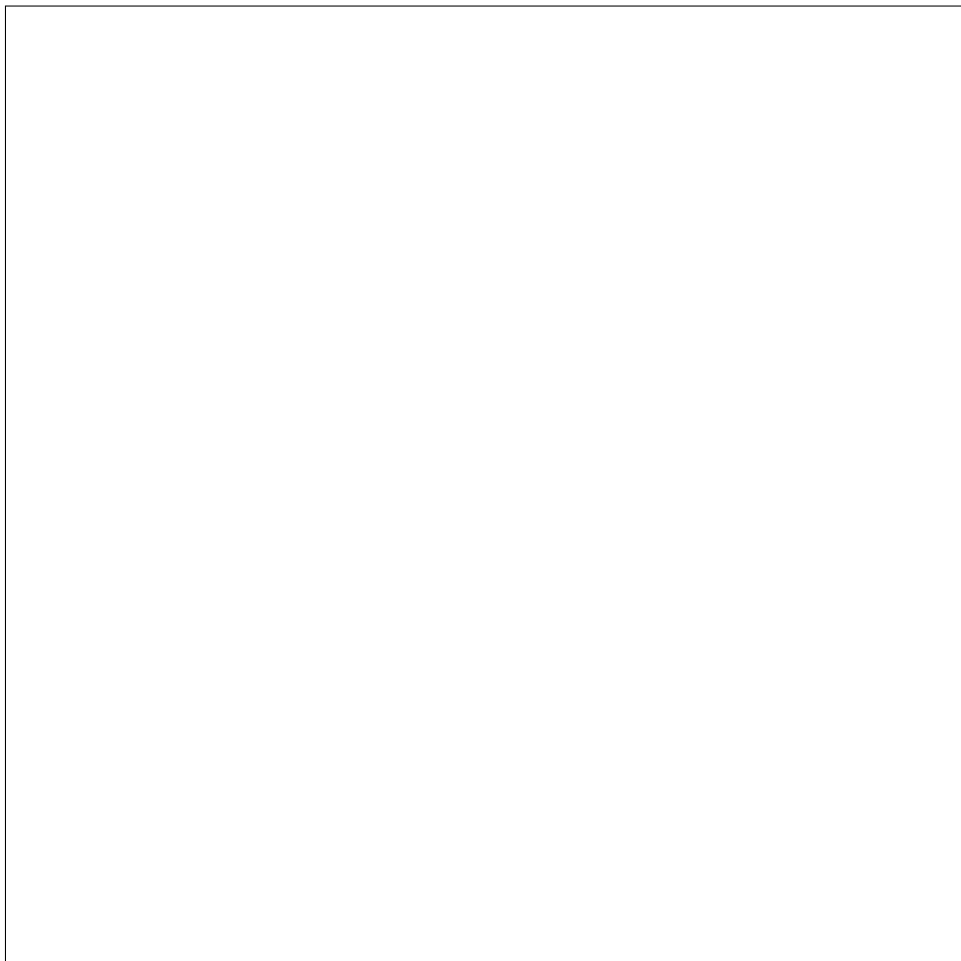
DVD E.1: Bronkode, simulاسies en allerlei



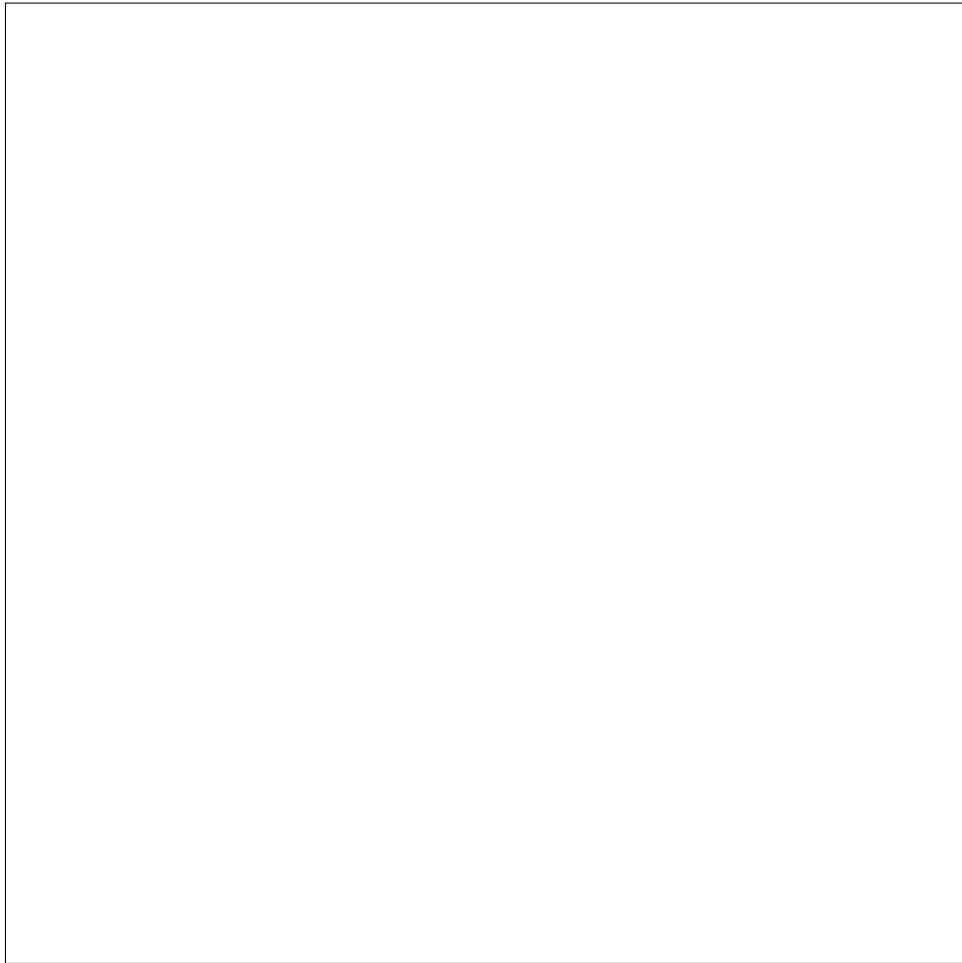
DVD E.2: Toetsdata en -kode 1



DVD E.3: Toetsdata en -kode 2



DVD E.4: Foto's en video's



DVD E.5: SD-kaart-afbeeldings

Bibliografie

- Adler, M. (2016 Oktober). pigz (weergawe 2.3.4). Beskikbaar by <https://github.com/madler/pigz> [2016-10-01].
- Audio Technica (2016). *AT8010: Omnidirectional Condenser Microphone*. Audio Technica.
- Bancroft, S. (1985 Januarie). An algebraic solution of the gps equations. *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-21, nr. 1, pp. 56–59.
- Behringer (1999). *ECM8000 Technical Specifications: Measurement Microphone*. Behringer.
- Berg, J.K. (1983). Vocalizations and associated behaviors of the African elephant (*Loxodonta Africana*) in captivity. *Ethology*, vol. 63, nr. 1, pp. 63–79.
- Brokaw, P. (2000). An IC amplifier user's guide to decoupling, grounding, and making things go right for a change. Application Note 202, Analog Devices. Beskikbaar by: www.analog.com/media/en/technical-documentation/application-notes/AN-202.pdf [2017-08-23].
- Byker, R. (2015). Die opsporing van olifantdreungeluide in raserige omgewings. Ongepubliseerde skripsie, Universiteit Stellenbosch.
- Byker, R. (2017 Maart). Antwoord op vraag: How do I make a device tree overlay for just a single GPIO? Beskikbaar by <https://raspberrypi.stackexchange.com/a/66548/41428> [2017-09-27].
- CDE Cornell Dubilier (2017). Multilayer ceramic capacitors: Performance characteristics. Beskikbaar by: <http://www.cde.com/resources/catalogs/ceramperf.pdf> [2017-06-23].
- Clemins, P.J., Trawicki, M.B., Adi, K., Tao, J. en Johnson, M.T. (2006). Generalized perceptual features for vocalization analysis across multiple species. In: *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, pp. 253–256. IEEE.
- Dooley, W. (2006). The perils of phantom power. Beskikbaar: <http://www.fixitinthemic.com/post/140710810134/the-perils-of-phantom-power> [2017-07-31].
- Drepper, U. *et al.* (2017 September). mman.h (deel van glibc, die GNU C-implementasie). Beskikbaar by <https://sourceware.org/git/?p=glibc.git;a=blob;f=misc/sys/mman.h;hb=HEAD> [2017-09-26].
- Duckitt, W. (2008 Maart). *The design of a high-performance, floating-point embedded system for speech recognition and audio research purposes*. Magistertesis, Universiteit Stellenbosch.
- Earthworks (datum onbekend). *M50: Premium High Definition Measurement Microphone*. Earthworks.
- Eggert, L., Eggert, J. en Woodruff, D. (2003). Estimating population sizes for elusive animals: the forest elephants of Kakum National Park, Ghana. *Molecular ecology*, vol. 12, nr. 6, pp. 1389–1402.

- Elephants Alive (2017). Vision and mission. Beskikbaar by: <http://elephantsalive.org/about-us/vision-and-mission/> [2017-06-13].
- Elwell, P. (2016 Maart). Kommentaar op GitHub: “Raspberry Pi 3 model B serial console does not use correct baudrate”. Beskikbaar by <https://github.com/raspberrypi/firmware/issues/553#issuecomment-199486644> [2016-03-21].
- Fang, B.T. (1990). Simple solutions for hyperbolic and related position fixes. *IEEE transactions on aerospace and electronic systems*, vol. 26, nr. 5, pp. 748–753.
- Frigo, M. en Johnson, S.G. (2005). The design and implementation of FFTW3. *Proceedings of the IEEE*, vol. 93, nr. 2, pp. 216–231. Spesiale uitgawe genaamd “Program Generation, Optimization, and Platform Adaptation”.
- Frigo, M. en Johnson, S.G. (2017 Oktober). FFTW: Fastest fourier transform in the west (weergawe 3.3.7). Beskikbaar by <http://www.fftw.org/> [2017-11-15].
- Garstang, M., Larom, D., Raspet, R. en Lindeque, M. (1995). Atmospheric controls on elephant communication. *Journal of Experimental Biology*, vol. 198, nr. 4, pp. 939–951.
- Georg Neumann GmbH (2015). What is self-noise (or equivalent noise level)? Beskikbaar by: <http://www.neumann.com/homestudio/en/what-is-self-noise-or-equivalent-noise-level> [2017-07-19].
- Gomersall, H. *et al.* (2016 Junie). pyFFTW: A pythonic python wrapper around FFTW (weergawe 0.10.4). Beskikbaar by <https://github.com/pyFFTW/pyFFTW> [2017-11-15].
- Henderson, G. (2017 September). Wiring Pi GPIO interface library for the Raspberry Pi. Beskikbaar by <http://wiringpi.com/> [2017-09-22].
- Hylands, D. (2012 Julie). CBUF: Circular buffer (hersiëning 20ae3d2). Beskikbaar by <https://github.com/dhylands/TimerUART/blob/master/CBUF.h> [2017-11-04].
- Invensense Inc. (2013 Desember). Recommendations for mounting and connecting invensense MEMS microphones. Application Note 1003, Invensense Inc. Beskikbaar by: <https://www.invensense.com/download-pdf/an-1003-recommendations-for-mounting-and-connecting-invensense-mems-microphones/> [2018-04-11].
- Invensense Inc. (2014). Recommendations for sealing InvenSense bottom-port MEMS microphones from dust and liquid ingress. Application Note 1124, Invensense Inc. Beskikbaar by: <https://www.invensense.com/download-pdf/an-1124-recommendations-for-sealing-invensense-bottom-port-mems-microphones-from-dust-and-liquid-ingress/> [2018-04-11].
- Invensense Inc. (2015). *ICS-40300: High SPL Analog Microphone with Extended Low Frequency Response*. Invensense Inc.
- Langbauer, W.R., Payne, K.B., Charif, R.A., Rapaport, L. en Osborn, F. (1991). African elephants respond to distant playbacks of low-frequency conspecific calls. *Journal of Experimental Biology*, vol. 157, nr. 1, pp. 35–46.
- Langbauer, W.R., Payne, K.B., Charif, R.A. en Thomas, E.M. (1989). Responses of captive African elephants to playback of low-frequency calls. *Canadian Journal of Zoology*, vol. 67, nr. 10, pp. 2604–2607.
- Larom, D., Garstang, M., Lindeque, M., Raspet, R., Zunckel, M., Hong, Y., Brassel, K., O’Beirne, S. en Sokolic, F. (1997a). Meteorology and elephant infrasound at Etosha National Park, Namibia. *The Journal of the Acoustical Society of America*, vol. 101, nr. 3, pp. 1710–1717.

- Larom, D., Garstang, M., Payne, K., Raspet, R. en Lindeque, M. (1997b). The influence of surface atmospheric conditions on the range and area reached by animal vocalizations. *Journal of experimental biology*, vol. 200, nr. 3, pp. 421–431.
- Lengauer, T. (2012). *Combinatorial algorithms for integrated circuit layout*. Springer Science & Business Media.
- Leong, K., Ortolani, A., Burks, K., Mellen, J. en Savage, A. (2003). Quantifying acoustic and temporal characteristics of vocalizations for a group of captive African elephants *loxodonta africana*. *Bioacoustics*, vol. 13, nr. 3, pp. 213–231.
- McCauley, M. (2016 Februarie). bcm2835: C library for Broadcom BCM 2835 as used in Raspberry Pi (weergawe 1.50). Beskikbaar by <http://www.airspayce.com/mikem/bcm2835/> [2016-02-28].
- McComb, K., Reby, D., Baker, L., Moss, C. en Sayialel, S. (2003). Long-distance communication of acoustic cues to social identity in african elephants. *Animal Behaviour*, vol. 65, nr. 2, pp. 317–329.
- Microchip (2011a). *MCP3903 ADC Evaluation Board for 16-Bit MCUs User's Guide*. Microchip.
- Microchip (2011b). *MCP3903: Six Channel Delta Sigma A/D Converter*. Microchip.
- Microchip (2011c). *PIC32MX3XX/4XX Data Sheet: High-Performance, General Purpose and USB, 32-bit Flash Microcontrollers*. Microchip.
- Microchip (2011d). 500 kHz, 800 μ A *Instrumentation Amplifier*. Microchip.
- Microchip (2013). *PIC32 family reference manual: Section 31. DMA Controller*. Microchip. Beskikbaar by: <http://ww1.microchip.com/downloads/en/DeviceDoc/60001117H.pdf> [2017-09-18].
- Microchip (2016). *PIC16(L)F1847: 18/20/28-Pin Flash Microcontrollers with XLP Technology*. Microchip. Beskikbaar by: <http://ww1.microchip.com/downloads/en/DeviceDoc/40001453F.pdf> [2017-09-18].
- Miller, G.E. en Raymond, E.S. (2016 September). *GPSD Time Service HOWTO*. Beskikbaar by <http://catb.org/gpsd/gpsd-time-service-howto.html> [2017-09-26].
- Mills, D.L. (2014 Maarta). *The Network Time Protocol Distribution*. Beskikbaar by <https://www.eecis.udel.edu/~mills/ntp/html/index.html> [2017-09-27].
- Mills, D.L. (2014 Augustusb). *Shared Memory Driver. (Deel van: The Network Time Protocol Distribution)*. Beskikbaar by <https://www.eecis.udel.edu/~mills/ntp/html/drivers/driver28.html> [2017-09-27].
- Minsky, H. (2009 April). Introduction to Reed Solomon codes. Beskikbaar by <http://rscode.sourceforge.net/rs.html> [2017-09-19].
- Minsky, H. (2013 April). A Reed-Solomon error-correcting encoder/decoder library written in C (weergawe 1.3). Beskikbaar by <https://sourceforge.net/projects/rscode> [2017-12-01].
- Mogul, J., Mills, D., Brittenson, J., Stone, J. en Windl, U. (2000 Maart). Pulse-Per-Second API for UNIX-like operating systems, version 1.0. RFC 2783, Network Working Group. Beskikbaar by <https://tools.ietf.org/html/rfc2783> [2017-09-27].

- NCO (2016 September). Selective availability. Tegn. Versl., National Coordination Office for Space-Based Positioning, Navigation, and Timing. Beskikbaar by <http://www.gps.gov/systems/gps/modernization/sa/> [2017-09-28].
- Network Time Foundation (2017 Maart). NTP: The network time protocol (weergawe 4.2.8p10). Beskikbaar by <http://www.ntp.org/> [2017-03-21].
- Nocedal, J. en Wright, S.J. (2006). *Numerical optimization*. Springer series in operations research, 2de uitg. Springer, New York. ISBN 0387303030.
- Ottoy, G. en De Strycker, L. (2016). An improved 2d triangulation algorithm for use with linear arrays. *IEEE Sensors Journal*, vol. 16, nr. 23, pp. 8238–8243.
- Payne, K.B., Langbauer, W.R. en Thomas, E.M. (1986). Infrasonic calls of the Asian elephant (*elephas maximus*). *Behavioral Ecology and Sociobiology*, vol. 18, nr. 4, pp. 297–301.
- Pbal, D. (2017 Junie). Mercury GPS library. Beskikbaar by https://github.com/Darthpbal/Mercury_Gps_Library [2017-09-28].
- Poole, J.H., Payne, K., Langbauer, W.R. en Moss, C.J. (1988). The social contexts of some very low frequency calls of African elephants. *Behavioral Ecology and Sociobiology*, vol. 22, nr. 6, pp. 385–392.
- Pornin, T. (2010 Maart). Antwoord op vraag: “Is fwrite non-blocking?”. Beskikbaar by <https://stackoverflow.com/a/2493713/836995> [2018-05-01].
- QANTEK Technology Corporation (datum onbekend). *QCP9 Series: 4.6x12.5 4-Pad Plastic SMD Crystal Unit*. QANTEK Technology Corporation.
- Raspberry Pi Foundation (2017 Septembara). *config.txt*. Beskikbaar by <https://www.raspberrypi.org/documentation/configuration/config-txt/README.md> [2017-09-23].
- Raspberry Pi Foundation (2017 Septemberb). *Device Trees, overlays, and parameters*. Beskikbaar by <https://www.raspberrypi.org/documentation/configuration/device-tree.md> [2017-09-21].
- Raspberry Pi Foundation (2017 Septemberc). *The Kernel Command Line*. Beskikbaar by <https://www.raspberrypi.org/documentation/configuration/cmdline-txt.md> [2017-09-21].
- Raspberry Pi Foundation (2017 Novemberd). *Memory options in config.txt*. Beskikbaar by <https://www.raspberrypi.org/documentation/configuration/config-txt/memory.md> [2017-11-09].
- Raspberry Pi Foundation (2017 Septembere). *Schematics*. Beskikbaar by <https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/README.md> [2017-09-27].
- Raspberry Pi Foundation (2017 Septemberf). *The Raspberry Pi UARTs*. Beskikbaar by <https://www.raspberrypi.org/documentation/configuration/uart.md> [2017-09-25].
- Raymond, E.S., Kuethe, C. en Miller, G. (2017 Septembara). *gpsd: interface daemon for GPS receivers*.
- Raymond, E.S. *et al.* (2016 Januarie). *gpsd: a GPS service daemon* (weergawe 3.16). Beskikbaar by <http://www.catb.org/gpsd/> [2016-01-08].
- Raymond, E.S. *et al.* (2017 Septemberb). *GPSPD Frequently Asked Questions*. Beskikbaar by <http://www.catb.org/gpsd/faq.html> [2017-09-28].

- Reed, I.S. en Solomon, G. (1960). Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, vol. 8, nr. 2, pp. 300–304.
- Robitaille, T. (2017 Oktober). fast-histogram: Fast 1d and 2d histogram functions in python (weergawe 0.3). Beskikbaar by <https://github.com/astrofrog/fast-histogram> [2017-11-15].
- Santiapillai, C., Wijeyamohan, S., Bandara, G., Athurupana, R., Dissanayake, N. en Read, B. (2010). An assessment of the human-elephant conflict in Sri Lanka. *Ceylon Journal of Science (Biological Sciences)*, vol. 39, nr. 1.
- Schoeman, L. (2017 Mei 18). Olifante weggehou by watergate. Die Burger.
- Schrijver, A. (2005). On the history of combinatorial optimization (till 1960). *Handbooks in operations research and management science*, vol. 12, pp. 1–68.
- SkyTraq Technology, Inc. (2010 Februarie). NMEA messages of SkyTraq Venus 6 GPS receiver. Application Note 0014, SkyTraq Technology, Inc. Beskikbaar by https://www.multitech.net/developer/wp-content/uploads/2010/02/AN0014_v02.pdf [2017-09-29].
- SkyTraq Technology, Inc. (2011 Oktobere). Binary messages of SkyTraq Venus 6 GPS receiver. Application Note 0003, SkyTraq Technology, Inc. Beskikbaar by https://www.sparkfun.com/datasheets/GPS/Modules/AN0003_v1.4.14_FlashOnly.pdf [2017-09-29].
- SkyTraq Technology, Inc. (2011 Junie). SkyTraq GPS viewer / configuration software (weergawe 0.4.833). Beskikbaar by http://cdn.sparkfun.com/datasheets/Sensors/GPS/Venus/638/viewer/GPS%20Viewer%20-%20Customer%20Release_110613.zip [2017-09-28].
- SkyTraq Technology, Inc. (2011 Januarie). *Venus638FLPx GPS Receiver Data Sheet*. SkyTraq Technology, Inc. Beskikbaar by https://cdn.sparkfun.com/datasheets/Sensors/GPS/Venus/638/doc/Venus638FLPx_DS_v07.pdf [2017-09-28].
- SkyTraq Technology, Inc. (2013 Maart). Position pinning extension for Venus 6 GPS receiver. Application Note 0025, SkyTraq Technology, Inc. Beskikbaar by <https://drive.google.com/file/d/0B901Zo8L0mjTMFE5ajZ3Y1h5eTVRekt1SUJIUGQ3Vn1PVFRR/view> [2017-09-29].
- Soltis, J., Leong, K. en Savage, A. (2005). African elephant vocal communication II: rumble variation reflects the individual identity and emotional state of callers. *Animal Behaviour*, vol. 70, nr. 3, pp. 589–599.
- Stoeger-Horwath, A.S., Stoeger, S., Schwammer, H.M. en Kratochvil, H. (2007). Call repertoire of infant african elephants: first insights into the early vocal ontogeny. *The Journal of the Acoustical Society of America*, vol. 121, nr. 6, pp. 3922–3931.
- Swift, J. (1726 Oktober). *Gulliver's travels*. Benjamin Motte. Beskikbaar by https://en.wikisource.org/wiki/Gulliver%27s_Travels [2017-09-25].
- Taylor, D. (2017 September). The Raspberry Pi as a stratum-1 NTP server. Beskikbaar by <http://www.satsignal.eu/ntp/Raspberry-Pi-NTP.html> [2017-09-27].
- Texas Instruments (2002). Analysis of the sallen-key architecture. Beskikbaar by: <http://www.ti.com.cn/cn/lit/an/sloa024b/sloa024b.pdf> [2017-08-14].
- Texas Instruments (2003 Desember). *PGA2500: Digitally controlled microphone preamplifier*. Texas Instruments. Beskikbaar by: <http://www.ti.com/lit/ds/symlink/pga2500.pdf> [2017-11-02].

- Texas Instruments (2012). *Rail-to-rail output CMOS operational amplifiers with shutdown*. Texas Instruments. Beskikbaar by: <http://www.ti.com/product/LMV344> [2018-05-01].
- Texas Instruments (2014). *TPS56x209, 4.5V to 17 V Input, 2-A, 3-A Synchronous Step-Down Voltage Regulator in 6 pin SOT-23*. Texas Instruments.
- Texas Instruments (2015). *TPS4021x 4.5-V to 52-V Input Current Mode Boost Controller*. Texas Instruments.
- Torvalds, L. *et al.* (2017 Augustusa). *The kernel's command-line parameters*. (Deel van: *The Linux Kernel documentation*). Beskikbaar by <https://github.com/torvalds/linux/blob/master/Documentation/admin-guide/kernel-parameters.rst> [2017-09-21].
- Torvalds, L. *et al.* (2017 Junieb). *NO_HZ: Reducing Scheduling-Clock Ticks*. (Deel van: *The Linux Kernel documentation*). Beskikbaar by https://github.com/torvalds/linux/blob/master/Documentation/timers/NO_HZ.txt [2017-09-21].
- Torvalds, L. *et al.* (2017 Januariec). *sched.h*. Deel van Linux-kern. Bronkode by <https://github.com/raspberrypi/linux/blob/rpi-4.4.y/include/linux/sched.h> [2017-01-06].
- Trønnes, N. (2016 Maart). *gpio-poweroff module*. Beskikbaar by <https://github.com/raspberrypi/linux/blob/rpi-4.4.y/arch/arm/boot/dts/overlays/gpio-poweroff-overlay.dts> [2017-09-27].
- u-blox (2009 Julie). *GPS antennas: RF design considerations for u-blox GPS receivers*. Application Note GPS-X-08014, u-blox. Beskikbaar by: [https://www.u-blox.com/sites/default/files/products/documents/GPS-Antenna_AppNote_\(GPS-X-08014\).pdf](https://www.u-blox.com/sites/default/files/products/documents/GPS-Antenna_AppNote_(GPS-X-08014).pdf) [2017-09-29].
- Venter, P.J. en Hanekom, J.J. (2010). Automatic detection of African elephant (*Loxodonta africana*) infrasonic vocalisations from recordings. *Biosystems engineering*, vol. 106, nr. 3, pp. 286–294.
- Verskeie outeurs (2016 Desember). *scipy.optimize.minimize: Minimization of scalar function of one or more variables*. (weergawe 0.19.0). Beskikbaar by <https://docs.scipy.org/doc/scipy-0.19.0/reference/generated/scipy.optimize.minimize.html> [2017-11-14].
- Wagner, D. *et al.* (2014 Januarie). *HOWTO: Build an RT-application*. Beskikbaar by https://rt.wiki.kernel.org/index.php/HOWTO:_Build_an_RT-application [2018-05-01].
- Whyte, I.J. (2001). Headaches and heartaches: the elephant management dilemma. *Environmental Ethics*, pp. 293–305.
Beskikbaar by: https://www.sanparks.org/docs/parks_kruger/elephants/more/head_heartaches.pdf
- Wijayakulasooriya, J.V. (2011). Automatic recognition of elephant infrasound calls using formant analysis and hidden markov model. In: *Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on*, pp. 244–248. IEEE.
- Windl, U., Dalton, D., Martinec, M. en Worley, D.R. (2006 November). *The NTP FAQ and HOWTO: Understanding and using the Network Time Protocol*. Beskikbaar by <http://www.ntp.org/ntpfaq/> [2017-09-27].
- Wood, A. (2017 Junie). *Pipe Viewer* (weergawe 1.6.6). Beskikbaar by <http://www.ivarch.com/programs/pv.shtml> [2017-06-30].
- Woolf, C. (2001). *Microphone Data*. Human Computer Interface. ISBN 095393540X.

- Zaca, M.N. (2015 Junie). South African co-ordinate systems. Beskikbaar by http://www.durban.gov.za/City_Services/engineering%20unit/Surveying_Land_Information/Documents/SouthAfricanCoordinateSystems.pdf [2017-09-29].
- Zeppelzauer, M., Hensman, S. en Stoeger, A.S. (2014). Towards an automated acoustic detection system for free-ranging elephants. *Bioacoustics*, vol. 24, nr. 1, pp. 13–29.
- Zeppelzauer, M., Stöger, A.S. en Breiteneder, C. (2013). Acoustic detection of elephant presence in noisy environments. In: *Proceedings of the 2nd ACM international workshop on Multimedia analysis for ecological data*, pp. 3–8. ACM.
- Zhang, J., Zhang, K., Grenfell, R. en Deakin, R. (2006). On the relativistic Doppler effect for precise velocity determination using GPS. *Journal of Geodesy*, vol. 80, nr. 2, pp. 104–110.
- Zhang, L. en Wu, X. (2006). On the application of cross correlation function to subsample discrete time delay estimation. *Digital Signal Processing*, vol. 16, nr. 6, pp. 682–694.